

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ННК «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Кафедра системного проектування
(повна назва кафедри)

«На правах рукопису»
УДК 004.852

«До захисту допущено»

Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)

“ _____ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 8.05010103 Системне проектування
(код і назва спеціальності)

на тему: Методи та моделі прийняття колективних рішень в мультиагентних системах

Виконав: студент 6 курсу, групи ДА-61м
(шифр групи)

_____ Севідов Павло Миколайович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник _____ проф., д.т.н., Рогоза В.С. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Розроблення стартап-проекту проф., д.т.н., Рогоза В.С. _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Інститут/факультет ННК «Інститут прикладного системного аналізу»
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною (освітньо-науковою) програмою

Спеціальність (спеціалізація) 8.05010103 Системне проектування
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.І. Петренко
(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Севідову Павлу Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Методи та моделі прийняття колективних рішень в мультиагентних системах»

науковий керівник дисертації Рогоза В.С., д.т.н., проф.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження Мультиагентні системи

4. Предмет дослідження Методи та моделі прийняття колективних рішень, які можливо використати в мультиагентних системах

5. Перелік завдань, які потрібно розробити провести огляд існуючих методів та технологій побудови мультиагентних систем, розробити метод або модель прийняття колективних рішень, провести моделювання та аналіз роботи метода або

моделі в мультиагентній системі, оформити роботу на основі отриманих результатів

6. Орієнтовний перелік публікацій Севідов П.М. Прийняття колективних рішень в мультиагентних системах / П. М. Севідов. // Міжнародний науковий журнал "Інтернаука". – 2018. – №7. – С. 81–101.

7. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Реалізація стартап-проекту			

8. Дата видачі завдання 01.02.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.02.2018	
2	Збір інформації та аналіз літератури	15.02.2018	
3	Проведення огляду існуючих методів побудови мультиагентних систем	28.02.2018	
4	Формалізація задачі розробки методу прийняття колективного рішення в мультиагентних системах	11.03.2018	
5	Реалізація поставленої задачі	13.04.2018	
6	Аналіз та порівняння результатів моделювання	25.04.2018	
7	Оформлення дипломної роботи	30.04.2018	
8	Отримання допуску до захисту та подача роботи в ДЕК	09.05.2018	

Студент

(підпис)

Севідов П.М.
(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Рогоза В.С.
(ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника

РЕФЕРАТ НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему: **Методи та моделі прийняття колективних рішень в
мультіагентних системах**

студентом: **Севідовим Павлом Миколайовичем**

Робота виконана на 82 сторінках, містить 13 ілюстрацій, 25 таблиць. При підготовці використовувалась література з 68 джерел.

Актуальність теми

Фундаментальним завданням багатьох системних мультіагентних мереж є успішне прийняття колективних рішень серед альтернатив за допомогою інформації, що поширюється по всій мережі. Групи окремих агентів, у програмах, включаючи транспортні та мобільні системи зондування, енерго та синтетичні біологічні мережі, часто вимагають єдиного вибору серед альтернатив, вибір істинного варіанта, як вчиняти та обрати необхідний керунок, або у випадку змін в середовищі та системі.

Тому дослідження даної технології, а також побудова нових моделей, що базуються на ній, є актуальним напрямком досліджень саме на сьогоднішній день, у час, коли все швидко змінюється, а єдиного підходу до прийняття колективних рішень в мультіагентних системах не існує.

Мета та задачі дослідження

Метою даної роботи є дослідження мультіагентних систем, а саме прийняття колективних рішень агентами в мультіагентних системах. Результатом проведених досліджень є практична частина роботи, що становить собою створення моделі для успішного прийняття рішення серед двох альтернатив та її апробація з використанням сучасних програмних засобів.

Рішення поставлених завдань та досягнуті результати

У даній роботі було представлено загальну агентну модель для біодинаміки колективного прийняття рішень, а також редукцію моделі та асимптотичне

розширення, і показано, як модель фіксує адаптивні та стійкі ознаки динаміки у прийнятті рішень серед медоносних бджіл. Так як медоносні бджоли надійно вибирають у користь найбільшого гнізда, а в разі альтернатив рівної цінності вони швидко роблять довільний вибір, якщо значення є досить високим.

Агентно-орієнтована модель прийняття рішень, про яку йдеться у даній роботі, пов'язує динаміку груп тварин та мультиагентних мереж. Вона утворює загальну структуру, що забезпечує досягнення характерних якостей прийняття рішень в колективі тварин (надійність, адаптивність) в інженерно-технічних мережевих системах. У найпростішому випадку "загальна комунікація" фіксує чутливість до прийняття рішень у перелітних птахів та медоносних бджіл. Попередні результати свідчать, що можливо зафіксувати демократичну динаміку консенсусу в зграях риб.

Об'єкт досліджень

Мультиагентні системи.

Предмет досліджень

Методи та моделі прийняття колективних рішень, які можливо використати в мультиагентних системах.

Методи досліджень

Для вирішення проблеми в даній роботі використовуються методи аналізу і синтезу, системного аналізу, порівняння, логічного узагальнення результатів.

Наукова новизна

Наукова новизна роботи полягає у створенні нових моделей для вирішення задач прийняття колективних рішень на прикладі як це роблять бджоли, застосовуючи мультиагентні системи як основний інструментарій для роботи в постійно змінюваному середовищі.

Практичне значення одержаних результатів

Отримані результати можуть використовуватись у майбутніх дослідженнях за напрямком покращення запропонованої моделі, враховуючи переваги та

недоліки даних результатів. Також дану модель можуть бути використані для покращення результатів роботи існуючих мультиагентних систем.

Публікації

Севідов П.М. Прийняття колективних рішень в мультиагентних системах / П. М. Севідов. // Міжнародний науковий журнал "Інтернаука". – 2018. – №7. – С. 81–101.

Ключові слова

Мультиагентні системи, колективне прийняття рішень, нелінійна динаміка, контроль біфуркації, мультиагентні системи.

ABSTRACT ON MASTER'S THESIS

on topic: Methods and models for making collective decisions in multi-agent systems

student: Pavlo M Sevidov

Work carried out on 82 pages containing 13 figures, 25 tables. The paper was written with references to 68 different sources.

Topicality

The fundamental task of many multi-agent systems is the successful adoption of collective decision making among alternatives through information distributed throughout the network. The groups of individual agents in the programs, including transport and mobile sensing systems, energy and synthetic biological networks, often require a single choice among alternatives, the choice of the true option, how to make and choose the required steering, or in the event of changes in the environment and system.

Therefore, the study of this technology, as well as the construction of new models based on it, is an actual direction of research to date, at a time when everything is changing rapidly, and there is no single approach to making collective solutions in multi-agent systems.

Purpose

The purpose of this work is to study multi-agent systems, namely, the adoption of collective decision making in multi-agent systems. The result of the research is the practical part of the work, which is the creation of a model for a successful decision making among the two alternatives and its testing using modern software.

Solution

In this paper a general agent model for biodynamics of collective decision-making was presented, as well as model reduction and asymptotic expansion, and it is shown how the model captures adaptive and stable signs of dynamics in making decisions among honeybees. Since honey bees reliably choose the largest nest, and in the case of

alternatives of equal value, they quickly make an arbitrary choice if the value is high enough.

The agent-oriented decision-making model mentioned in this paper links the dynamics of animal groups and multi-agent networks. It forms a general structure that ensures the achievement of the qualities of decision-making in a team of animals (reliability, adaptability) in engineering and network systems. In the simplest case, "general communication" captures sensitivity to decision making in migratory birds and honey bees. The preliminary results indicate that it is possible to capture the democratic dynamics of consensus in a series of fish.

Object of research

Multi-agent systems.

Subject of research

Methods and models of collective decision making that can be applied in multi-agent systems.

Research methods

To solve the problem in this paper were used methods of analysis and synthesis, system analysis, comparison, logical generalization of results.

Scientific novelty

The scientific novelty of the work is to create new models for solving the problems of making collective decisions on the example how bees do it, using multi-agent systems as the main tools for working in a constantly changing environment.

The practical value of the results

The results obtained can be used in future studies to improve the proposed model, taking into account the advantages and disadvantages of these results. Also, this model can be used to improve the performance of existing multi-agent systems.

Publications

Sevidov P.M. Collective decision making in multiagent systems / P.M. Sevidov. // International scientific magazine "Internet Science". – 2s018 - №7. - P. 81-101.

Keywords

Multiagent systems, collective decision making, nonlinear dynamics, bifurcation control, multi-agent systems.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	12
ВСТУП.....	13
1 ІНТЕЛЕКТУАЛЬНІ АГЕНТИ: МУЛЬТИАГЕНТНІ СИСТЕМИ	15
1.1 Мульти-Агентні Системи	17
1.2 Агентно-орієнтоване моделювання та симуляція	21
1.3 Агентно-орієнтоване програмне забезпечення.....	24
1.4 Висновки до розділу.....	27
2 ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ ПОБУДОВИ МУЛЬТИАГЕНТНОЇ СИСТЕМИ.....	29
2.1 Побудова мультиагентної системи за допомогою SPADE.....	29
2.2 Побудова мультиагентної системи за допомогою JADE.....	31
2.3 Побудова мультиагентної системи за допомогою JADEX.....	32
2.4 Побудова мультиагентної системи за допомогою AnyLogic	33
2.5 Технологічний процес розробки мультиагентної системи.....	34
2.5.1 Використання систем керування версіями для побудови мультиагентних систем.....	35
2.5.2 Використання систем керування проектом та задачами для побудови мультиагентних систем	38
2.6 Висновки до розділу.....	42
3 ПРИЙНЯТТЯ КОЛЕКТИВНИХ РІШЕНЬ АГЕНТАМИ В МУЛЬТИАГЕНТНИХ СИСТЕМАХ	44
3.1 Модель прийняття рішень на основі агентів.....	45
3.2 Бджолиний вулик та значення значеннєвого прийняття рішень	48
3.3 Зниження моделі до низького рівня, прийнятної множини	52

	11
3.4 Отримування сумісності та вплив системних параметрів	54
3.4.1 Наближення точки біфуркації для симетричної, всеосязної мережі Z_2	55
3.4.2 Відновлення значення динаміки медоносних бджіл	57
3.4.3 Вплив параметрів системи.	57
3.5 Висновки до розділу	59
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ «МУЛЬТИАГЕНТНА СИСТЕМА ДЛЯ ПРИЙНЯТТЯ КОЛЕКТИВНИХ РІШЕНЬ АГЕНТАМИ»	60
4.1 Опис ідеї та технологічної аудит стартап-проекту	60
4.2 Аналіз ринкових можливостей	62
4.3 Розробка ринкової стратегії проекту	69
4.4 Розробка маркетингової програми	71
4.6 Висновки до розділу	74
ВИСНОВКИ	76
ПЕРЕЛІК ПОСИЛАНЬ	78

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

FIPA - Foundation for Physical Intelligent Agents

ІІІ – штучний інтелект

МАС – мультиагентна система

АОМС - Агентно-орієнтоване моделювання та симуляція

XML – eXtensible Markup Language

ВСТУП

Основоположним завданням багатьох мультиагентних систем є вдале прийняття колективних рішень серед наявних альтернатив за допомогою інформації, що розповсюджується по всьому середовищу. Як вже було зазначено в публікації: “Групи окремих агентів, у програмах, включаючи транспортні та мобільні системи зондування, енерго та синтетичні біологічні мережі, часто вимагають єдиного вибору серед альтернатив, вибір істинного варіанта, як вчиняти та обрати необхідний керунок, або у випадку змін в середовищі та системі”. Іншим прикладом є створення нового бізнесу, для якого необхідно максимально правильно оцінити його потенціальну прибутковість, визначити оптимальну ринкову нішу спрогнозувати його роботу в різних умовах ринкового середовища. Розв’язання задачі прогнозування процесів життєвого циклу підприємства на тому чи іншому етапі розвитку, у більшості випадків пов’язане з моделюванням залежностей результуючих характеристик від різноманітних вхідних факторів. Вони можуть мати різноманітну природу та визначатися як ситуацією на ринку, так і внутрішніми параметрами підприємства, в той же час динаміка зміни цих факторів також має велике значення, що значно ускладнює розрахунок потенційної ефективності компанії традиційними аналітичними методами, результатом якого має бути прийняте однозначне рішення вигідності відкриття бізнесу для поточного стану середовища.

З метою розробки розподілених мультиагентних рішень прагнемо використовувати механізми, що використовуються групами тварин, виживання яких залежить від успішних колективних рішень серед наявних альтернатив. Домашні вуликові бджоли [66], зграйна риба [53] та перелітні птахи [54] приймають ефективні рішення, незалежно від конфліктів чи значних змін у середовищі. Бджоли використовують децентралізований підхід та стикаються з обмеженням відчуттів,

комунікації та обчислення [60, 68], проте вони все ще діють із сталою швидкістю розвитку, точністю, надійністю та адаптивністю [64].

Для вирішення цього завдання буде розглянуто загальну агентно-орієнтовану динамічну модель прийняття рішень розподіленого між двома альтернативами. У цьому типі прийняття рішень виборча біфуркація є повсюдною [61]; це виявляється, наприклад, у динаміці прийняття рішень домашніх медоносних бджіл та зграї золотого синця між джерелами поживи. Підхід полягає у введенні моделі агента так, щоб вона також містила біфуркацію типу вилка. Це дозволяє тісно пов'язати динаміку групи тварин та мультиагентну динаміку шляхом відображення на звичайну форму біфуркації вилки.

Основний внесок цієї роботи полягає в наступному. По-перше, буде представлено загальну агентну модель для біодинаміки колективного прийняття рішень, а також редукцію моделі та асимптотичне розширення, щоб показати, як модель фіксує адаптивні та стійкі ознаки динаміки у прийнятті рішень серед медоносних бджіл. Примітно, що медоносні бджоли надійно вибирають у користь найбільшого гнізда, а в разі альтернатив рівної цінності вони швидко роблять довільний вибір, якщо значення є досить високим. Дана динаміка медоносів була досліджена в роботах [63, 65, 66, 67].

По-друге, досліджено як значення альтернатив, індивідуальних переваг та топології взаємодії впливає на динаміку прийняття рішень. Це мотивовано проблемою проектування колективної динаміки прийняття рішень, оскільки ці показники можуть служити контрольними параметрами в інженерно-технічних системах [3].

1 ІНТЕЛЕКТУАЛЬНІ АГЕНТИ: МУЛЬТИАГЕНТНІ СИСТЕМИ

Агент - це метафора, яка спочатку виникла в "Штучному інтелекті" (ШІ) [17]. Його характеристики змінювалися, коли акцент робився на мультиагентних системах, а не на окремих агентах. Ключовою подією в історії мультиагентних систем стало визнання того, що агентів можна використовувати для моделювання та впровадження розподілених систем. Створення Foundation for Physical Intelligent Agents (FIPA) походить від визнання того, що мультиагентні системи є новим та перспективним підходом для впровадження та розгортання розподілених систем. Важливо визнати, що часто використовують агентів поза межами ШІ, тому що інструменти та методи, які мають призначення для одного, не обов'язково є корисними для іншого. Зокрема, в даний час мультиагентні системи мають два основних додатків за межами своєї традиційної ролі в ШІ:

1. Проектування розподілених систем з впровадженням агентно-орієнтованої програмної інженерії (AOSE) (наприклад, агенти-орієнтовані методології, агенти-орієнтовані мови програмування).
2. Моделювання та симуляція на основі агента (ABMS), про який йдеться в основному в цій роботі.

Агенти - це суб'єкти, які спостерігають за своїм оточенням та діють на нього, щоб досягти своїх цілей [1, 17]. Дві фундаментальні характеристики агентів - це *автономія* та *ситуативність*. Автономія означає, що агенти мають розумний цикл, який дає їм контроль над їх внутрішнім станом і поведінкою. Враховуючи те, що агенти розташовані, тому що вони можуть "відчувати", сприймати та управляти середовищем, в якому вони працюють. Навколишнє середовище може бути фізичним або віртуальним, і це розуміється агентами з точки зору (релевантних) даних. Автономія передбачає активність, тобто здатність агента вживати заходів для досягнення своїх цілей, не вимагаючи цього.

З точки зору програмування, агент-орієнтоване програмування було введено Шохамом як "спеціалізоване об'єктно-орієнтоване програмування" [23]. Різниця між агентами та статичними об'єктами зрозуміла. Посилаючись на Wooldridge [1]: (1) об'єкти не контролюють власну поведінку (це підсумовується добре відомим девізом "Об'єкти роблять це безкоштовно; агенти роблять це тому, що вони хочуть"), (2) об'єкти не демонструють гнучкості у своїй поведінці, і (3) у стандартних моделях об'єктів існує один потік керування, а агенти за своєю суттю мають багато поточкових властивостей.

Агентно-орієнтована парадигма також відрізняється від моделі Актора (Actor Model) [19] (і від активних об'єктів (Active Objects), значною мірою натхненних останнім). Насправді, актори не мають ні цілей, ні мети, навіть якщо їх специфікація включає в себе процес. Агенти замість цього експлуатують свій обґрунтований цикл (як контрольний потік), можливо, разом з ключовими абстракціями віри, бажання та наміру (як логіку), з тим щоб реалізувати алгоритми, наприклад, процеси дій у своєму оточенні для здійснення їх цілі. Іншими словами, об'єкти "роблять це" безкоштовно, оскільки вони є даними, агенти - це процеси та "роблять це", оскільки вони є функціональними для їх завдань.

Навколишнє середовище, в якому розташовані агенти, не виявляє такої автономії, яка є типовою агентам, хоча вона може розвиватися, також завдяки внутрішньому процесу. Однак його діяльність не спрямована на досягнення мети, і це робить середовища більш схожими на активні об'єкти.

Біноміальне агент-середовище формалізується за допомогою моделювання підходів, таких як [20], де середовище розглядається як забезпечення "навколишніх умов для існування агентів і що опосередковує як взаємодію між агентами, так і доступ до ресурсів", і зокрема за допомогою мета-моделі агент і артефакт [23].

Система, що складається з декількох взаємодіючих агентів, називається мультиагентною системою. На цьому рівні широко визнається, що подальші абстракції стають в нагоді, як організації та взаємодії, спрямованих на сприяння

змістовному та плідному координуванню автономних та неоднорідних агентів у системі. Таким чином, агенти розташовуються не тільки в фізичному середовищі, вони також знаходяться в соціальному середовищі, де вони вступають у взаємодію з іншими агентами та підпадають під дію правил середовища, до якого вони належать. Нормативна мультиагентна система - це "мультиагентна система разом з нормативними системами, в яких агенти, з одного боку, можуть вирішити, чи слід дотримуватися явно представлених норм, а з іншого - нормативні системи визначають, як і в якій мірі агенти можуть змінювати норми" [22]. Вплив на обґрунтований цикл агента полягає в тому, що агенти можуть обґрунтувати соціальні наслідки своїх дій.

1.1 Мульти-Агентні Системи

Немає єдиного визначення слова-агента, і не існує єдиного визначення для терміна мультиагентної системи (МАС). Примітно, що основні прийняті визначення визначають спільні риси, такі як взаємодія агентів у системі: через спільне середовище, через структуровані повідомлення (онтології, протоколи взаємодії). Дійсно, МАС можна визначити з точки зору взаємодіючих сутностей, зокрема агентів. Зв'язок може відрізнятися від простих форм до складних. Проста форма зв'язку полягає в тому, що обмежується простими сигналами з фіксованими інтерпретаціями. Такий підхід був використаний Джордом у плануванні мультиагентів, щоб уникнути конфліктів, коли план був синтезований кількома агентами. Більш складна форма зв'язку - за допомогою структури робочої області. Робоча область - загальний ресурс, зазвичай розділений на декілька областей, залежно від різних типів знань або різних рівнів абстракції у вирішенні проблем, за допомогою яких агенти можуть читати або записувати відповідну інформацію для своїх дій. Іншою формою зв'язку є повідомлення, що проходить між агентами.

Автономія є ще однією основною характеристикою агентів при визначенні МАС, також називається "самоорганізовані системи", що дозволяє їм знаходити найкраще рішення своїх проблем "без втручання". Головною особливістю, яка

досягається при розробці мультиагентних систем, є гнучкість, адже мультиагентна система може бути додана, модифікована та реконструйована, без необхідності детального переписування програми. МАС також має тенденцію запобігати розповсюдженню розломів, самостійно відновлюватися та бути відмовостійкою, в основному через нерелевантність компонентів.

Дуже важливо розрізняти *Автоматичні* та *Автономні* системи. Автоматичні системи повністю попередньо запрограмовані і діють повторно і незалежно від зовнішнього впливу або управління. Вона може бути описана як самостійна або саморегульована, і вона здатна стежити за зовнішнім заданим шляхом, компенсуючи невеликі відхилення, викликані зовнішніми розривами. Однак вона не може визначити шлях відповідно до певної мети або вибрати мету, що диктує його шлях. Враховуючи те, що автономні системи, як МАС, самостійно спрямовуються на мету, оскільки вони не вимагають зовнішнього контролю, а скоріше, вони керуються законами та стратегіями, які чітко відрізняють традиційні та мультиагентні системи. Якщо використовуються методи машинного навчання, автономні системи можуть розробляти гнучкі стратегії для себе, за допомогою яких вони вибирають свою поведінку.

Більш детально, норми є фундаментальним компонентом мультиагентних систем, які керують очікуваною поведінкою щодо конкретної ситуації. За допомогою норм подано бажану поведінку для населення природної або штучної спільноти. Дійсно, вони, як правило, розуміються як правила, що вказують на очікувані дії, які мають бути обов'язковими, заборонними або допустимими, виходячи з певного набору фактів. За даними Холландра та Ву [24], норми були використані для позначення обмежень на поведінку [25], щоб створити рішення для проблеми макрорівні [26] і служити обов'язковим [27], регулюючі або контрольні пристрої для децентралізованих систем [28]. Найбільш поширені норми:

1. Правила, які є природними нормами, що виникають без будь-якого застосування [29]. Правила вирішують проблеми координації, коли немає

конфлікту між особистістю та колективними інтересами; наприклад, кожен відповідає бажаній поведінці. Основні норми використовуються для вирішення або полегшення проблем колективної дії, коли виникає конфлікт між особистістю та колективними інтересами [28, 30]. Наприклад, норма не забруднювати міські вулиці має важливе значення, оскільки вона вимагає від людей транспортувати їх сміття, а не розпоряджатися ним на місці, актом, який вигідний кожному.

2. Нормативні норми. Нормативні норми призначені для регулювання діяльності шляхом встановлення обов'язку чи заборони при виконанні дії.
3. Конституційні норми, які затверджуються для вироблення нових цілей норм або станів справ, наприклад, правила гри, як шахи.
4. Процедурні норми, які класифікуються як об'єктивні та суб'єктивні. Об'єктивні процесуальні норми являють собою правила, що виражають, як рішення дійсно виробляються в нормативній системі, тоді як суб'єктивні процесуальні норми являють собою інструмент для осіб, які працюють у системі, наприклад, процедури зворотного зв'язку.

Координація - ще один відмінний фактор МАС. По суті, агент існує і здійснює свою діяльність у середовищі, в який входять інші агенти. Тому координація дій між агентами має важливе значення для досягнення цілей та послідовної роботи. Координація передбачає розглянути дії інших агентів у системі при плануванні та виконанні дій одного агента. Координація дозволяє агентам досягти узгодженої поведінки всієї системи. Координація може означати співпрацю, і в цьому випадку середовище агентів працює над досягненням спільних цілей, але може також означати конкуренцію, агентам, що мають розбіжності чи навіть антагоністичні цілі. У цьому останньому випадку координація важлива, оскільки агент повинен враховувати дії інших, наприклад, конкуруючи за певний ресурс або пропонуючи той же сервіс.

Ще однією характеристикою МАС є його виняткова поведінка. Поведінка, що розвивається в агентах, зазвичай визначається як поведінка, яка не приписується жодному окремому агенту, але є глобальним результатом координації агентів [29]. Це визначення підкреслює, що поведінка, що виникає, є колективною поведінкою. Є також інші визначення. Поведінка, що виникає, - це те, що неможливо прогнозувати за допомогою аналізу на будь-якому рівні простіше, ніж системою в цілому. Вихідна поведінка, за визначенням, залишається після того, як все інше було прояснено [31]. Це визначення висвітлює складність у прогнозуванні та поясненні поведінки, що виникає. Якщо поведінка є передбачуваною та зрозумілою, то вона не буде розглядатися як непередбачувана поведінка, і підходи можуть бути розроблені для обробки поведінки. Виникнення також визначається як дію простих правил, які об'єднують для отримання складних результатів [27]. Це визначення говорить, що правила, що застосовуються до осіб, можуть бути досить простими, але колективне поведінка групи може виявитися досить складною і непередбачуваною. Дослідники розробили експерименти, щоб продемонструвати таку ситуацію. Хоча це правда, що вся поведінка походить від окремих людей, взаємодії - це те, що ускладнює розуміння речей. Похідне поведінки - це, по суті, будь-яка поведінка системи, яка не є властивістю будь-якого компонента цієї системи, і виникає через взаємодії між компонентами системи. Запозичення з біологічних моделей, таких як колонія мурашок, евакуаційна поведінка також може розглядатися як виробництво високого рівня або складної поведінки через взаємодію декількох простих правил. Деякі приклади взаємодій, що виникають: поведінка колонії бджіл, де колективний збір нектару оптимізується за допомогою танцювального руху окремих робочих бджіл; Стадність птахів не можна описати поведінкою окремих птахів; збої ринку не можна пояснити "підсумовуванням" поведінки окремих інвесторів.

Додаткові подробиці про мультиагентні системи можуть бути знайдені в роботі [1].

1.2 АГЕНТНО-ОРІЄНТОВАНЕ МОДЕЛЮВАННЯ ТА СИМУЛЯЦІЯ

Агентно-орієнтоване моделювання та симуляція (АОМС) - це категорія обчислювальних моделей, що використовують динамічні дії, реакції та протоколи взаємодії серед агентів у спільній середовищі, щоб оцінити їхній дизайн та продуктивність, а також отримати уявлення про їх поведінку та властивостях [21]. Агенти та мультиагентні системи - це об'єкти, які можуть бути ефективно використані для моделювання складних систем, що складаються з взаємодіючих об'єктів. Саме тому вони були прийняті для вивчення біологічних та хімічних систем, особливо коли системи стали надто складними для аналітичних інструментів, доступних в хімії, фізиці та математичної фізики. Той факт, що агенти та мультиагентні системи є абстракціями з виконуваними партнерами, агенти та мультиагентні системам, які підтримують багато інструментів, сприяли використанню агентні технології для моделювання біологічних та хімічних систем. Розмір імітованих систем і високий рівень точності імітаційних явищ вимагає спеціальних інструментів, що дозволяють експерту області описувати моделювання з невеликим об'ємом знань або взагалі не вдаючись в інженерні питання, пов'язані з розподіленими системами. Агентно-орієнтовані технології забезпечують такі інструменти, і підтримують доменних експертів при побудові ефективних розподілених систем з мінімальним акцентом на питаннях розподілених систем. Примітно, що навіть якщо спеціальні інструменти доступні, загальноприйняті інструменти, призначені для допомоги для створення агенційно-орієнтованого програмного забезпечення в рамках АОМС. Зокрема, існує низка інструментів на основі агента для підтримки моделювання та моделювання складних та / або розподілених систем [23]:

1. Netlogo: це мультиагентне програмне середовище моделювання, яке дозволяє моделювати природні та соціальні явища. Це особливо добре підходить для моделювання складних систем з часом. Дійсно, інженери можуть надати інструкції сотням або тисячам "агентів", які працюють незалежно. Це дає

змогу вивчити зв'язок між поведінкою мікрорівнів окремих осіб та шаблонами макрорівня, що виникають внаслідок їх взаємодії. Вона поставляється з великою бібліотекою існуючих симуляцій, як спільних, так і традиційних, які можна використовувати та змінювати в різних галузях, таких як суспільствознавство та економіка, біологія та медицина, фізика та хімія, математика та інформатика. У традиційному моделюванні NetLogo, симуляція виконується відповідно до правил, зазначених автором симуляції. Ще однією особливістю NetLogo є HubNet, технологія, яка дозволяє використовувати NetLogo для запуску спільного моделювання. HubNet додає новий вимір до NetLogo, дозволяючи моделювання проводитись не тільки відповідно до правил, але й безпосередньої участі людини.

2. FLAME: це універсальна система моделювання агента, яка може використовуватися для розробки додатків у багатьох областях. Моделі створюються на основі моделі обчислень, що називаються (розширені) кінцеві автомати. Фреймворк може автоматично генерувати програми моделювання, які можуть ефективно запускати моделі на суперкомп'ютерах. Він випускає повні агентно-орієнтовані програми, які можна скопіювати та виконати на більшості комп'ютерних системах, починаючи від ноутбуків до суперкомп'ютерів. Крім того, FLAME забезпечує модельну бібліотеку, яка є набором відносно простих моделей, що ілюструють використання FLAME у різних програмах.
3. AnyLogic: це інструмент моделювання, який підтримує всі найпоширеніші методології моделювання на сьогоднішній день: системна динаміка, процес-орієнтована (дискретні події) та агентно-орієнтоване моделювання. Його візуальне середовище суттєво прискорює процес розробки. Він має моделі для таких сфер: логістики виробництва, поставки ланцюгів ринку, конкуренції бізнес-процесів, моделювання охорони здоров'я, моделювання фармацевтичних продуктів, руху пішохідних потоків, інформаційно-

телекомунікаційних мереж, моделювання соціальних процесів, керування активами маркетингу, фінансових операцій з моделюванням операцій та оптимізації плану.

4. Repast: Repast Suite - це сімейство передових, вільних і відкритих джерел агентно-орієнтованого моделювання та моделювання платформ, які спільно розвиваються протягом багатьох років. Repast Symphony - це інтерактивна та проста у вивченні основана на Java система моделювання, призначена для використання на робочих станціях та невеликих обчислювальних кластерах. Розширена версія називається Repast for High Performance Computing, яка є нерозвиненою та експертно-орієнтованою C++-орієнтованою системою, яка призначена для використання на великих обчислювальних кластерах та суперкомп'ютерах.
5. Jason: Це інтерпретатор для розширеної версії AgentSpeak, який був одним з найбільш впливових абстрактних мов на основі архітектури BDI. Jason реалізує операційну семантику цієї мови, сильне заперечення, тому доступні і загальноприйняті припущення, і відкритий світ. Анотації з віруваннями використовуються для інформації про мета рівня та анотації на мітках плану. Одним з найвідоміших підходів до розвитку когнітивних агентів є архітектура BDI (Beliefs-Desires-Intents). Це забезпечує можливість запуску мультиагентної системи, розподіленої через мережу.
6. Framsticks: це тривимірний проект моделювання життя. Моделюються як механічні структури (тіла), так і системи управління (мізки) істот. Можна розробляти різні види експериментів, включаючи просту оптимізацію, коеволюцію, відкриту та спонтанну еволюцію, різні генофонди та популяції, моделювання видів та екосистем. Користувачі цього програмного забезпечення працюють над еволюційними обчисленнями, штучним інтелектом, нейронними мережами, біологією, робототехнікою та

моделюванням, когнітивною наукою, нейронаукою, медициною, філософією, віртуальною реальністю, графікою та мистецтвом.

7. Gephi: це інтерактивна платформа візуалізації та розвідки для всіх видів мереж та складних систем, динамічних та (1) ієрархічних графіків. Це дозволяє (2) аналіз досліджуваних даних: інтуїтивно-орієнтований аналіз за допомогою мережевих маніпуляцій в режимі реального часу; Аналіз зв'язку: виявлення основних структур асоціацій між об'єктами, зокрема в мережах без масштабу; (3) Аналіз соціальних мереж: легке створення з'єднувачів соціальних даних для карток громадських організацій та мереж малого масштабу; Аналіз біологічної мережі: представлення моделей біологічних даних; (5) Створення плаката: пропаганда наукової роботи з друкованими картами високої якості. Він працює в основному з метрикою, що стосується центральності, ступеня (силове право), щільності, довжини шляху, діаметра, HITS, модульності, коефіцієнта кластеризації.
8. Stanford Network Analysis Platform (SNAP): це загальноприйнята, високопродуктивна система аналізу та маніпулювання великими мережами. Вона легко масштабується до масивних мереж із сотнями мільйонів вузлів і мільярдами мереж. Вона ефективно маніпулює великими графіками, обчислює структурні властивості, генерує регулярні та випадкові графіки та підтримує атрибути на вузлах та краях.

Це список найбільш популярних інструментів моделювання, які можуть бути використані для моделювання, імітації та аналізу складних систем шляхом прийняття агентно-орієнтованої парадигми.

1.3 АГЕНТНО-ОРІЄНТОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

З ранніх днів наприкінці 1960-х років, програмне забезпечення постійно стикається з проблемою кращого розуміння джерел складності програмних систем. Протягом останніх трьох десятиліть складність взаємодії між частинами поступово визначається як один із найважливіших джерел такої складності. Програмні

системи, що містять, можливо, велике число взаємодіючих частин є критичними, особливо коли діапазон взаємодій динамічно змінюється, частини мають свою власний вузол керування, а частини взаємодіють, керуючись складними протоколами [1]. Як наслідок, основною темою досліджень програмного забезпечення була розробка методів та інструментів для розуміння, моделювання та впровадження систем, в яких взаємодія є основним джерелом складності. Це призвело до пошуку нових обчислювальних абстракцій, моделей та інструментів для роз'яснення та впровадження МАС, які були визнані прототипними прикладами таких систем. Агентно-орієнтоване моделювання та симуляція (АОМС) - це нова парадигма програмного забезпечення, розроблена з метою наближення до складної складності аналізу та впровадження МАС [2]. Впродовж останніх двох десятиліть було запропоновано кілька методологій АОМС, і [13] наводяться недавні огляди сучасного стану таких методологій.

Інтерес до мов програмування для агентів походить від впровадження агентних технологій, і з того часу він швидко розвивався. Справді, мови програмування для агентів виявилися особливо зручними для моделювання та розробки складних МАС. Сьогодні мови програмування агентів є важливою темою дослідження, і вони широко визнаються як важливі інструменти для розробки агентних технологій, на відміну від основних мов, які часто вважаються непридатними для ефективного впровадження АОМС. Мови програмування для агентів зазвичай базуються на конкретних моделях агентів, і вони спрямовані на забезпечення специфічних конструкцій для прийняття таких моделей на високому рівні абстракції. Особливості різноманітних мов програмування для агентів, що пропонуються протягом багатьох років, можуть суттєво відрізнятися, наприклад, від обраного агентивного ментального ставлення (якщо є), інтеграції з платформою агента (якщо така існує), основною парадигмою програмування та базовою мовою реалізації. Деякі класифікації відповідних мов програмування для агента вже запропоновані для порівняння характеристик різних мов та для забезпечення чіткого

огляду стану можливостей. Bădică et al. [32] класифікує мови програмування для агентів на основі використання психічного ставлення. Відповідно до такої класифікації, мови програмування для агентів можна розділити на: мови агентно-орієнтованого програмування (АОП), мови Belief Desire Intentions (BDI), гібридні мови (які поєднують дві попередні класи) та інші мови, які виходять за межі попередніх класів. Варто зазначити, що така класифікація визнає, що мови BDI слідує за агенційно-орієнтованою парадигмою програмування, як це визначено в [33], але він приділяє особливу увагу їм за їх помітну актуальність у літературі. Bordini et al. [34] пропонує іншу класифікацію, де мови поділяються на декларативні, обов'язкові та гібридні. Декларативні мови є найбільш поширеними, тому що вони зосереджені на автоматичних міркуваннях, як з точки зору АОП, так і з точки зору BDI. Також були запропоновані деякі відповідні імперативні мови, і більшість з них були отримані шляхом додавання певних конструкцій до існуючих мов процедурних програм. Нарешті, наявність (або відсутність) основополагаючої мови є важливою основою порівняння мов програмування агента.

Навіть якщо ранні пропозиції починаються наприкінці 1990-х років, АОМС все ще знаходиться на ранній стадії еволюції. Хоча існує безліч гарних аргументів, які підтверджують думку про те, що агенти є важливим напрямком для розробки програмного забезпечення, існує ще потреба у реальному досвіді, щоб підкріпити ці аргументи. Методології та інструменти для підтримки розгортання МАС починають прийматись для критично-важливих додатків, але повільно. Незважаючи на те, що було запропоновано декілька методологій аналізу та розробки, орієнтованих на агенти, між ними існує порівняно незначний консенсус. У більшості випадків навіть не існує згоди щодо тих видів концепцій, які повинна підтримувати методологія. Але дослідження щодо АОМС все ще активні, і вони нещодавно були активізовані з точки зору АОМС з точки зору моделі, орієнтованого на розвиток, як підсумовується в Kardas [34].

1.4 ВИСНОВКИ ДО РОЗДІЛУ

Агенти та мультиагентні системи використовуються для вивчення та моделювання складних систем у різних областях застосування, де фізичний фактор присутній для мінімізації енергії, коли фізичні об'єкти, як правило, досягають найнижчого споживання енергії у фізично стримуваному світі. Крім того, МАС інтенсивно використовуються для аналізу шляхом моделювання біологічних та хімічних систем. Література повідомляє про різні успішні використання абстракцій та їх виконуваних інструментів. Зокрема, вивчення переваг та витрат на залучення агентів та мультиагентних систем для підтримки наукових досліджень біологічних та хімічних систем не було широко розглянуте.

Відома область застосування, де агенти та МАС були успішно експлуатовані, вивчення біологічних явищ, є синтез білків, загальне та актуальне явище в природі. У цій галузі кілька підходів для прогнозування тривимірної структури білків, наприклад, доступні в літературі Jennings et al. [32]. Деякі з них експлуатують хімічні чи фізичні властивості білків (наприклад, Standley et al. [47], Dudek та співавт. [25], робота з мінімізації енергії, тоді як Галактионов і Маршалл [28], Sabzekar et al. [40] використовує внутрішньоглобальні контакти); інші використовують еволюційну інформацію (Piccolbon and Mauri [37]). Деякі інструменти для прогнозування тривимірної структури білків були представлені в Douget and Labesse [24].

Це показує, що автономія агентів, їх нормована свобода взаємодії та можливість розгортання великомасштабних систем з мінімальним доглядом за проблемами, пов'язаними з розподіленими системами, є лише однією з основних переваг наближення моделювання та моделювання складних систем з агентами. Крім того, вони тісно відображають, як природні системи працюють, поширюючи проблему серед ряду реактивних, автономних, дорадчих, активних, адаптивних, можливо, мобільних, гнучких та спільних об'єктів. Всі ці властивості роблять технологію агента більш придатними, ніж інші (наприклад, розподілену систему,

штучну інтелектуальну систему) для вирішення проблем і високого рівня складності біологічної системи та пов'язаних з цим явищ.

2 ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ ПОБУДОВИ МУЛЬТИАГЕНТНОЇ СИСТЕМИ

Агентно-орієнтовані технології забезпечують такі інструменти, і підтримують доменних експертів при побудові ефективних розподілених систем з мінімальним акцентом на питаннях розподілених систем. Примітно, що навіть якщо спеціальні інструменти доступні, загальноприйняті інструменти, призначені для допомоги для створення агенційно-орієнтованого програмного забезпечення в рамках АОМС. Зокрема, існує низка інструментів на основі агента для підтримки моделювання та моделювання складних та / або розподілених систем [23].

Нижче проведено огляд найпопулярніших технологій побудови мультиагентних систем. Далі буде обрано порівняльний аналіз технологій та буде обрано технологію для проведення досліджень.

2.1 ПОБУДОВА МУЛЬТИАГЕНТНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ SPADE

SPADE (Smart Python multi-Agent Development Environment) - це мультиагентна платформа, заснована на технологіях XMPP/Jabber і написана в мові програмування Python. Ця технологія пропонує безліч функцій і можливостей, що полегшують створення МАС, таких як канал зв'язку, концепції користувачів (агентів) і серверів (платформ) і зручний протокол зв'язку на основі XML, подібний до FIPA-ACL. Існує безліч інших агентських платформ, але SPADE є першим, хто базується на технологіях XMPP [4].

Платформа SPADE Agent не вимагає (але наполегливо рекомендує) роботу агентів, створених за допомогою бібліотеки агентів SPADE. Сама платформа використовує бібліотеку для розширення можливостей її внутрішніх компонентів, але крім цього є можливість створювати власні агенти на вибраній вами мові програмування та використовувати їх з SPADE. Єдиною вимогою, є те що ці агенти мають підтримувати можливість зв'язку через протокол XMPP. Повідомлення FIPA-ACL будуть введені в повідомлення XMPP. Однак слід попереджати, що деякі

функції SPADE можуть бути недоступними, якщо ви не використовуєте бібліотеку агентів SPADE для створення ваших агентів. (рис. 2.1)

Основні характеристики SPADE:

1. Побудований на основі FIPA.
2. Реалізовано наступні MTP: XMPP, P2P, HTTP та SIMBA.
3. Має зручний інтерфейс для управління платформою з web.
4. Дозволяє групування агентів.

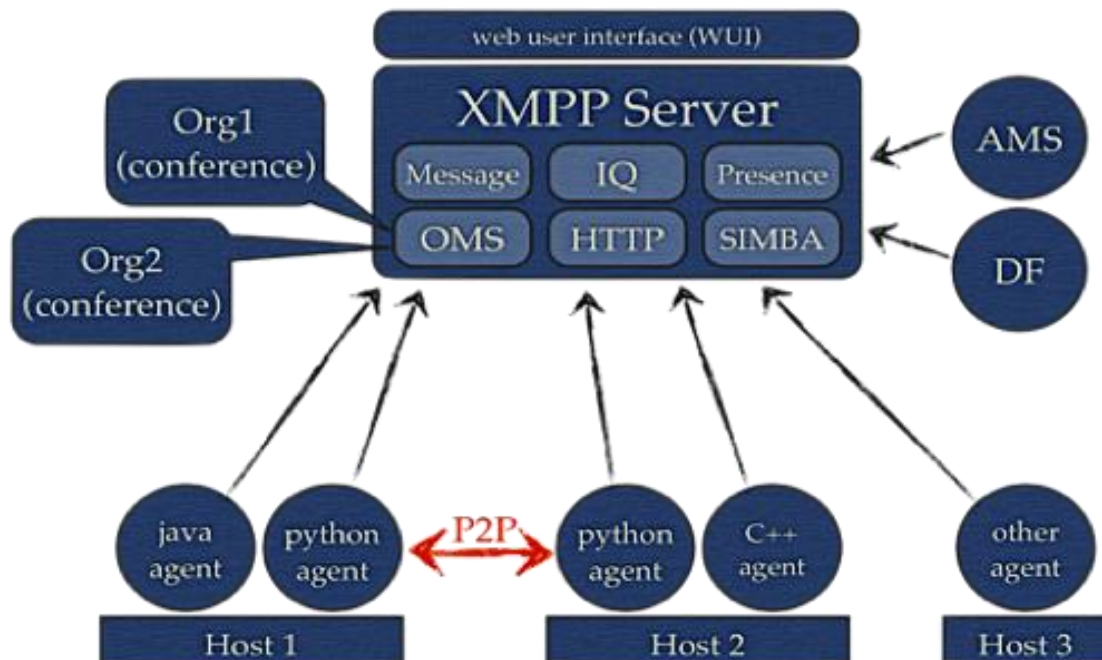


Рис. 2.1 – Складові частини SPADE-проекту[9]

2.2 ПОБУДОВА МУЛЬТИАГЕНТНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ JADE

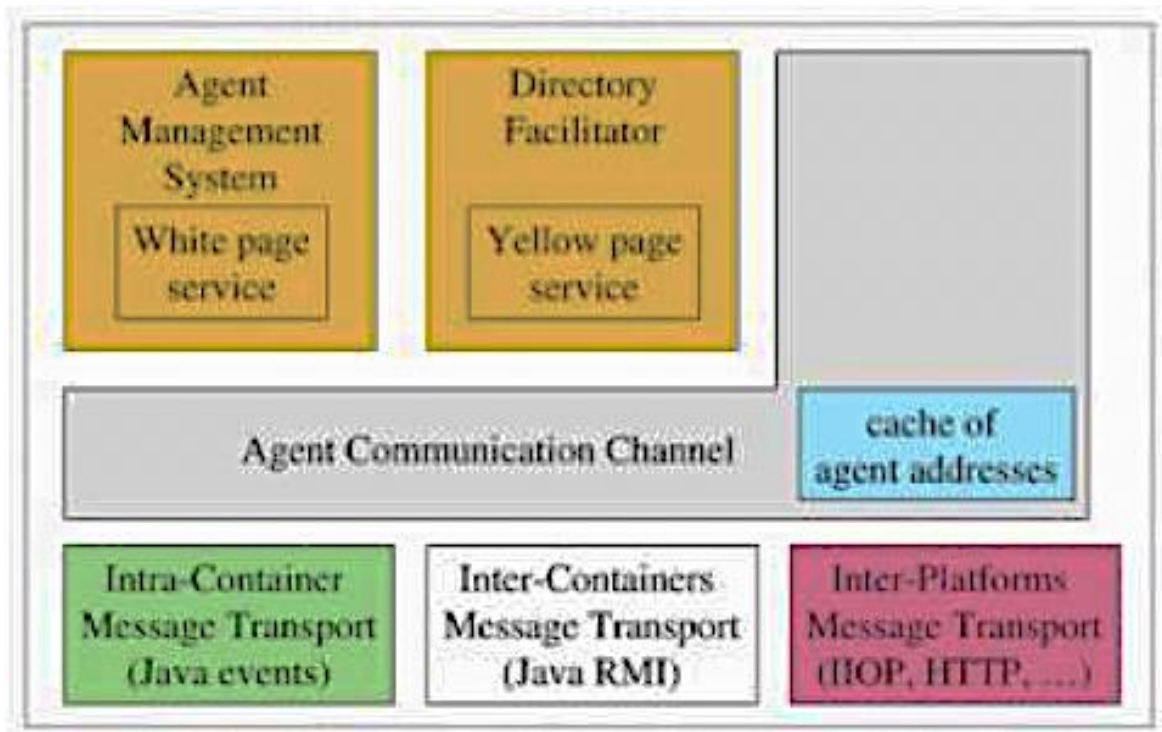


Рис. 2.2 – Складові частини JADE [10]

Java Agent Development Framework (рис. 2.2), або JADE, є програмним забезпеченням для розробки інтелектуального агента, реалізованого в Java. Також є проміжним програмним забезпеченням, яке полегшує розробку мультіагентних систем відповідно до стандарту FIPA, для чого створюється кілька контейнерів для агентів, кожна з яких може працювати на одній або декількох системах. Система JADE підтримує координацію між декількома агентами FIPA і забезпечує стандартну реалізацію мови спілкування FIPA-ACL, що полегшує комунікації між агентами. JADE був спочатку розроблений компанією Telecom Italia і поширюється як вільне програмне забезпечення під ліцензією GNU Lesser General Public License [10].

JADE пропонує такі можливості:

1. Середовище, де виконуються агенти JADE
2. Бібліотеку класів для створення агентів, що використовують наслідування та перевизначення поведінки

3. Графічний набір інструментів для моніторингу та управління платформою інтелектуальних агентів

Основним способом обміну зв'язку між агентами є відправка повідомлень `ACLMessage`. Повідомлення, що надсилає `ACL` (Language Communication Agent), є базою зв'язку між агентами. Відправлення повідомлень здійснюється методом відправки класу `Agent`. У цьому методі передається об'єкт типу `ACLMessage`, який містить інформацію одержувача, мову, кодування та вміст повідомлення. Ці повідомлення надсилаються асинхронно, тоді як повідомлення отримуються, вони будуть зберігатися в черзі повідомлень.

2.3 ПОБУДОВА МУЛЬТИАГЕНТНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ JADEX

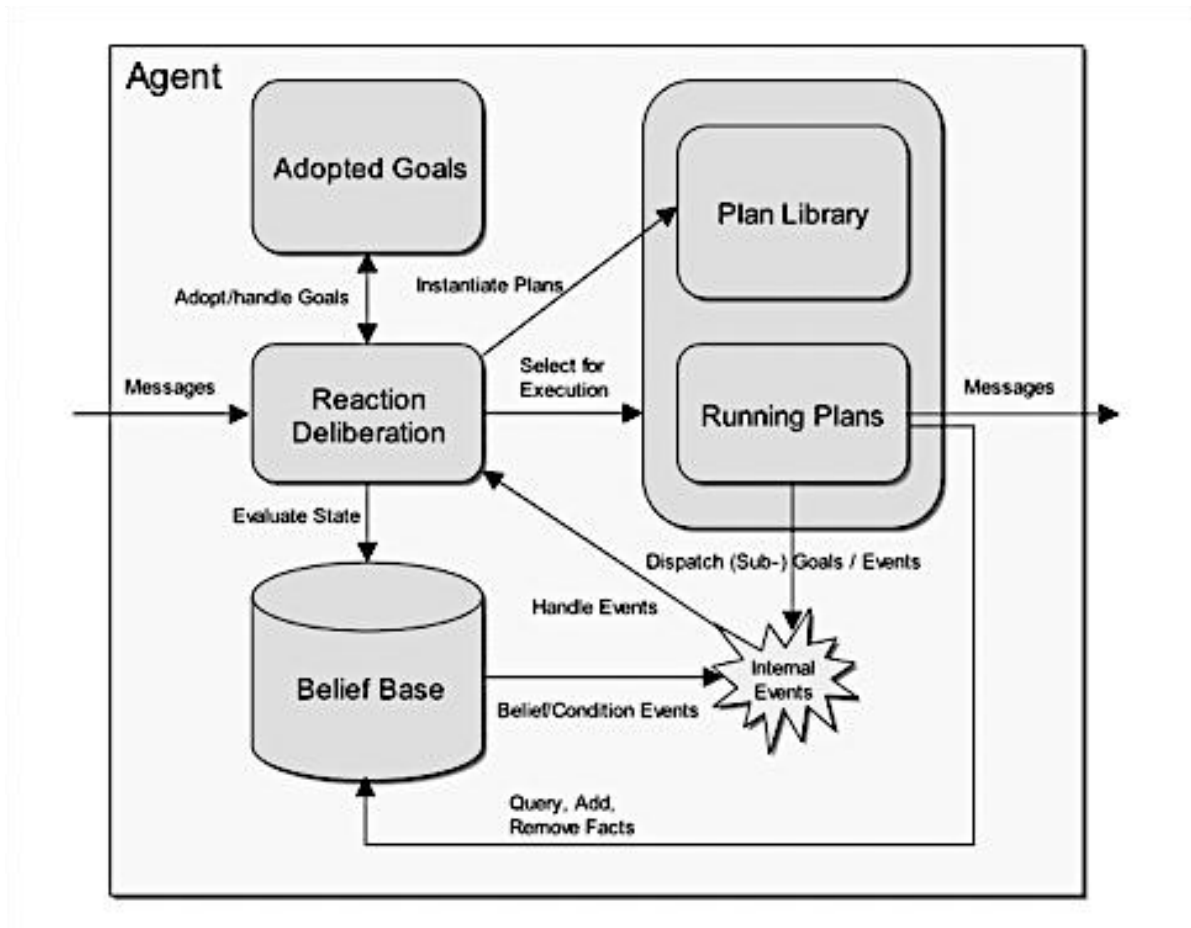


Рис. 2.3 – Структура проекту JADEX

Основною ідеєю при розробці JADEX був в створенні компромісу між програмуванням агентів на рівні проміжного програмного забезпечення та

логічними моделями. Для реалізації цього було вирішено використовувати вже існуючий програмне забезпечення – JADE, яке вже включає в собі комунікаційну інфраструктуру і сервіси керування платформою, такі як управління агентами, розробкою та роботою середовища в відповідності до FIPA.

Дане програмне забезпечення призначене для побудови логіко-орієнтованих моделей (повністю реалізовано зовнішню взаємодію агентів і управління ними, внутрішня поведінка програмується користувачем), тож є гарною основою для створення систем, основою яких є логічні моделі[11].

2.4 ПОБУДОВА МУЛЬТИАГЕНТНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ ANYLOGIC

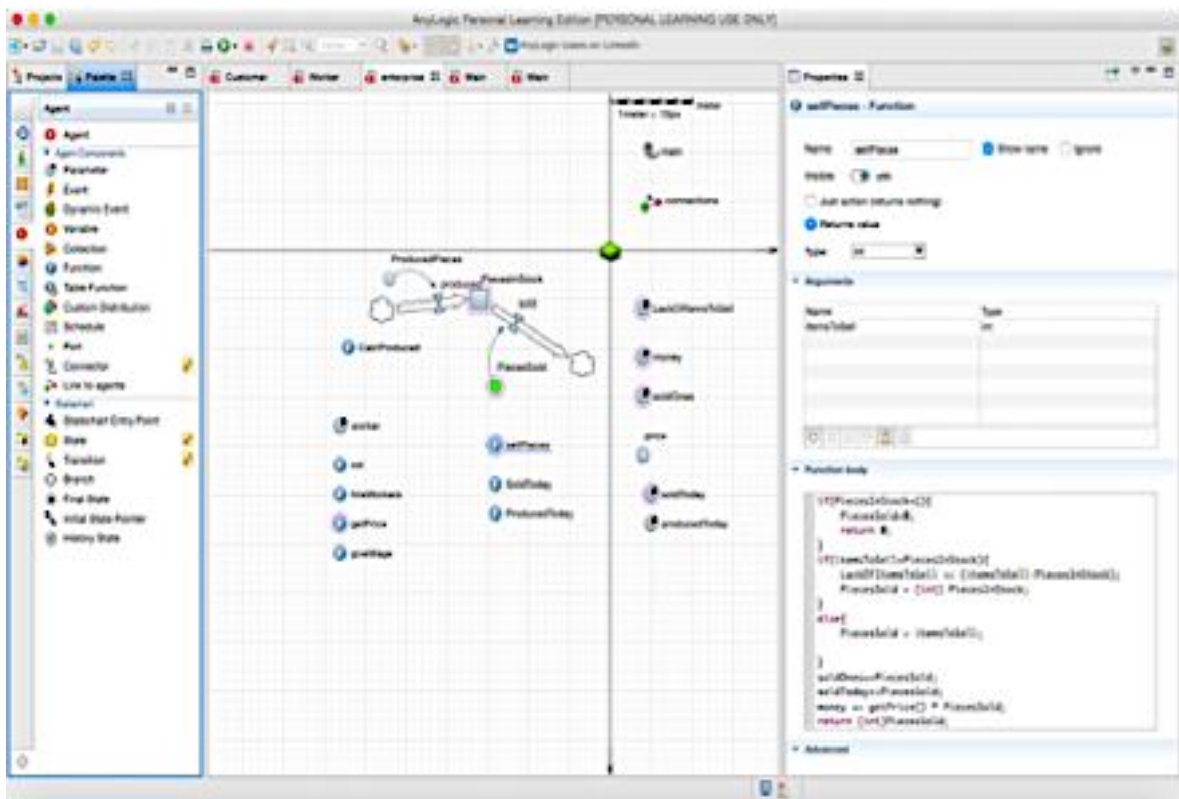


Рис. 2.4 – Приклад середовища AnyLogic

AnyLogic - програмне забезпечення для імітаційного моделювання. Інструмент обладнаний сучасним графічним інтерфейсом і дозволяє використовувати Java для розробки моделей.

Графічне середовище AnyLogic включає в себе наступні компоненти:

1. Stock & Flow Diagrams (діаграма потоків та накопичувачів) застосовується при розробці моделей, використовуючи метод системної динаміки.
2. Statecharts (карти станів) в основному використовується в агентних моделях для визначення поведінки агентів. Але також часто використовується в дискретно-подієвому моделюванні, наприклад для симуляції машинних збоїв.
3. Action charts (блок-схеми) використовується для побудови алгоритмів. Застосовується в дискретно-подієвому моделюванні (маршрутизація дзвінків) і агентному моделюванні (для логіки рішень агента).
4. Process flowcharts (діаграми процесів) основна конструкція, що застосовується для визначення процесів в дискретно-подійному моделюванні[12].

Середовище моделювання AnyLogic підтримує проектування, розробку, документування моделі, виконання комп'ютерних експериментів з моделлю, включаючи різні види аналізу - від аналізу чутливості до оптимізації параметрів моделі щодо деякого критерію[12].

Середовище розробки AnyLogic включає в себе наступний набір стандартних бібліотек:

1. Process Modeling Library – розроблена для підтримки дискретно-подієвого моделювання в таких областях як Виробництво, Ланцюги поставок, Логістика і Охорона здоров'я.
2. Pedestrian Library – створена для моделювання пішохідних потоків в «фізичної» навколишньому середовищу.
3. Rail Yard Library – підтримує моделювання, імітацію та візуалізацію операцій сортувальної станції будь-якої складності і масштабу.

2.5 ТЕХНОЛОГІЧНИЙ ПРОЦЕС РОЗРОБКИ МУЛЬТИАГЕНТНОЇ СИСТЕМИ

Виконання наукової дисертації має супроводжуватись власними дослідженнями, які мають мати можливість відтворитися на різних фізичних

машинах, тому було вирішено використати інфраструктуру управління проектами в хмарі.

2.5.1 ВИКОРИСТАННЯ СИСТЕМ КЕРУВАННЯ ВЕРСІЯМИ ДЛЯ ПОБУДОВИ МУЛЬТИАГЕНТНИХ СИСТЕМ

Система керування версіями - це управління змінами в документах, комп'ютерних програмах, великих веб-сайтах та інших колекціях інформації. Зміни, як правило, ідентифікуються за числовим чи буквеним кодом, які називаються "номером ревізії", "рівнем ревізії" або просто "ревізією". Наприклад, початковий набір файлів - "revision 1". Коли здійснюється перша зміна, результуючий набір - "revision 2" і так далі. Кожна ревізія пов'язана з позначкою часу та особою, яка вносить зміни. Зміни можна порівняти, відновлювати, а з деякими типами файлів об'єднати.

Система контролю дозволяє зберігати попередні версії файлів та завантажувати їх за потребою. Вона зберігає повну інформацію про версію кожного з файлів, а також повну структуру проекту на всіх стадіях розробки. Місце зберігання даних файлів називають репозиторієм. В середині кожного з репозиторіїв можуть бути створені паралельні лінії розробки — гілки.

Потреба у логічному способі організувати та контролювати зміни існувала практично настільки, наскільки існує письмо, але контроль над переглядами став набагато важливішим і складним, коли почалася ера обчислень. Нумерація книжкових видань та перегляду специфікацій - це приклади, що відносяться лише до періоду друку. Сьогодні найбільш функціональними (а також складними) системами контролю версій є ті, що використовуються у розробці програмного забезпечення, де команда людей може змінювати ті ж файли. Тому для продуктивної роботи над великим проектом, де працює немала кількість розробників, необхідно використовувати систем контролю версій.

Багато системи управління версіями надають ряд інших можливостей:

1. Дозволяють створювати різні варіанти одного документа, т.н. *гілки*, із загальною історією змін до точки розгалуження і з різними - після неї.
2. Дають можливість дізнатися, хто і коли додав або змінив конкретний набір рядків у файлі.
3. Ведуть журнал змін, в який користувачі можуть записувати пояснення про те, що і чому вони змінили в цій версії.
4. Контролюють права доступу користувачів, дозволяючи або забороняючи читання або зміна даних, в залежності від того, хто запитує цю дію.

Нижче розглянуто найбільш використовувані на даний момент системи контролю версій[5].

Subversion - централізована система (на відміну від розподілених систем, таких як Git або Mercurial), тобто дані зберігаються в єдиному сховищі. Сховище може розташовуватися на локальному диску або на мережевому сервері.

Робота в Subversion мало відрізняється від роботи в інших централізованих системах управління версіями. Клієнти копіюють файли зі сховища, створюючи локальні робочі копії, потім вносять зміни в робочі копії і фіксують ці зміни в сховище. Кілька клієнтів можуть одночасно звертатися до сховища. Для спільної роботи над файлами в Subversion переважно використовується модель копіювання - зміна - злиття. Крім того, для файлів, що не допускають злиття (різні бінарні формати файлів), можна використовувати модель блокування - зміна - розблокування.

При збереженні нових версій використовується дельта-компресія: система знаходить відмінності нової версії від попередньої і записує тільки їх, уникаючи дублювання даних.

При використанні доступу за допомогою WebDAV також підтримується прозоре управління версіями - якщо будь-який клієнт WebDAV відкриває для

запису і потім зберігає файл, що зберігається на мережевому ресурсі, то автоматично створюється нова версія [5].

Основними недоліками SVN є централізованість репозиторію, низька швидкість роботи, наявність помилок при змінах типу імені файлів та директорій і низька гнучкість роботи [5].

GIT – розподілена система контролю версій. Система спроектована як набір програм, спеціально розроблених з урахуванням їх використання в сценаріях. Це дозволяє зручно створювати спеціалізовані системи контролю версій на базі Git або призначені для користувача інтерфейси.

Git підтримує швидке поділ і злиття версій, включає інструменти для візуалізації та навігації по нелінійної історії розробки. Як і Mercurial, Git надає кожному розробнику локальну копію всієї історії розробки, зміни копіюються з одного сховища в інший.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH- або HTTP-сервером. TCP-сервіс git-daemon входить в дистрибутив Git і є поряд з SSH найбільш поширеним і надійним методом доступу. Метод доступу по HTTP, незважаючи на ряд обмежень, дуже популярний в контрольованих мережах, тому що дозволяє використовувати існуючі конфігурації мережевих фільтрів. основними перевагами якої є надзвичайно висока в порівнянні з SVN швидкість роботи, гнучкість, peer-to- peer обмін набором змін та зручність використання в великих проектах.

Проте існують певні недоліки – велика кількість “зайвих” опцій та складність робочого процесу, неефективна в малих проектах.

Mercurial — розподілена система керування версіями файлів та спільної роботи. Система Mercurial написана на Python, хоча чутливі до продуктивності частини (наприклад, своя реалізація diff) виконані в якості модулів-розширень на C. Mercurial спочатку була написана для Linux, пізніше перенесена під Windows,

MacOS X і більшість Unix-систем. Репозиторії Mercurial управляються за допомогою утиліти командного рядка hg, але є і GUI-інтерфейси.

Поряд з традиційними можливостями систем контролю версій, Mercurial підтримує повністю децентралізовану роботу (відсутнє поняття основного сховища коду), розгалуження (можливо вести кілька гілок одного проекту і копіювати зміни між гілками), злиття репозиторіїв (чим і досягається «розподіленість» роботи). Підтримується обмін даними між репозиторіями через HTTP/HTTPS, SSH і вручну за допомогою упакованих наборів змін.

Утиліта hg володіє компактним інтерфейсом, і Mercurial вважається більш простий в освоєнні системою, ніж, наприклад, git.

Так як дослідження проводяться однією людиною і одночасно необхідно не забувати про можливість масштабування було вирішено використовувати GIT як компроміс між Subversion та Mercurial.

2.5.2 ВИКОРИСТАННЯ СИСТЕМ КЕРУВАННЯ ПРОЕКТОМ ТА ЗАДАЧАМИ ДЛЯ ПОБУДОВИ МУЛЬТИАГЕНТНИХ СИСТЕМ

Контроль проекту — це елемент, який забезпечує відповідність проекту графіку виконання та бюджету. Контроль проекту починається з планування та закінчується звітом з виконання проекту, пронизуючи кожен елемент процесу управління проектом. Кожен проект має бути оцінений щодо рівня необхідного контролю: забагато контролю означає втрату часу, замало контролю означає збільшення ризиків. Якщо контроль проекту впроваджений не вірно, вартість для бізнесу пояснюється у термінах помилок, виправлень та додаткових витрат на аудит. Системи контролю необхідні для витрат, ризиків, якості, комунікацій, часу, змін, закупівель та людських ресурсів. До того ж аудитори мають визначити наскільки проекти впливають на фінансову звітність, наскільки достовірну інформацію отримують замовники і скільки точок контролю існує. Аудитори мають розглянути процес розробки та процедури на предмет способу впровадження. Процес виробництва та якість кінцевого продукту також може бути оцінений, якщо

така потреба виникає. Бізнес може забажати від аудиторської фірми фіксування проблем на ранніх етапах з метою зменшення зусиль необхідних на виправлення. Аудитор може виступати як консультант з контролю, частина проектної команди або як окремих аудитор, частина аудиту. Бізнес іноді використовує формалізовані процеси розробки систем. Це допомагає підтвердити успішність розробки. Формальний процес ефективніший у створенні сильних точок контролю. Аудитори мають перевірити такий процес та підтвердити його якісну організацію та відповідність практиці. Гарний формальний план впровадження системи характеризує:

1. Стратегія, задля приведення розробки до загальніших цілей організації.
2. Стандарти для нових систем.
3. Політики управління проектом щодо часу та бюджету.
4. Процедури, що описують процес.
5. Оцінка якості змін[6].

Спеціалізоване програмне забезпечення використовується для вирішення проблеми автоматизації управління проектом. Найбільш використовувані на даний момент системи це Redmine, YouTrack та Jira. Останні два (YouTrack та Jira) є платним програмним забезпеченням. Тому було прийнято рішення в сторону Redmine (рис. 2.5), який є безкоштовним.

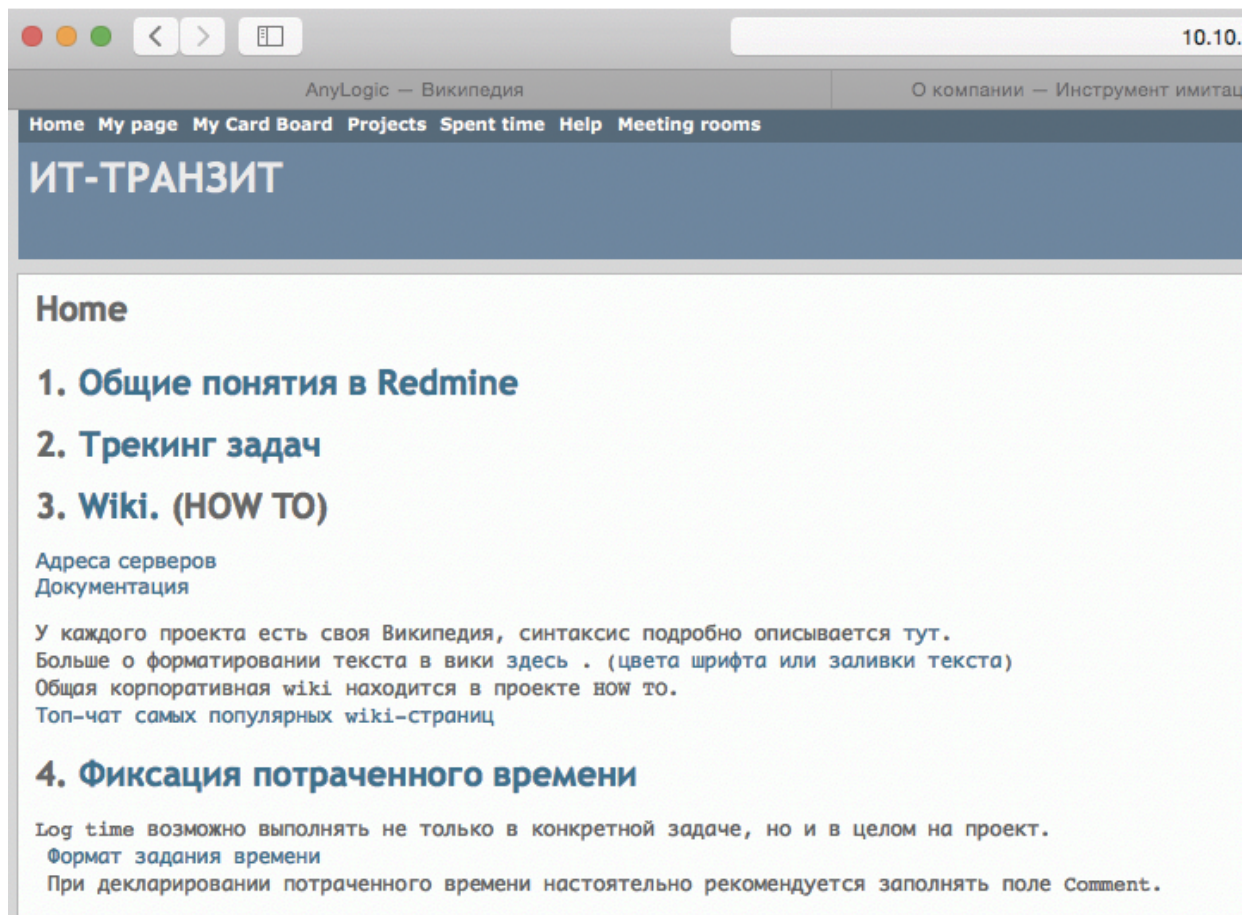


Рис 2.5 – Приклад роботи Redmine

Діаграма Ганта - це популярний тип стовпчастих діаграм (гістограм), які використовуються для ілюстрації плану, графіка робіт з якого-небудь проекту. Є одним з методів планування проектів. Використовується в додатках з управління проектами.

По суті, діаграма Ганта складається з смуг, орієнтованих уздовж осі часу. Кожна смуга на діаграмі представляє окреме завдання в складі проекту (вид роботи), її кінці - моменти початку і завершення роботи, її протяжність - тривалість роботи. Вертикальною віссю діаграми служить перелік завдань. Крім того, на діаграмі можуть бути відзначені сукупні завдання, відсотки завершення, показники послідовності і залежності робіт, мітки ключових моментів (віхи), мітка поточного моменту часу «Сьогодні» і інше.

Ключовим поняттям діаграми Ганта є «Віха» - мітка значимого моменту в ході виконання робіт, спільний кордон двох або більше завдань. Віхи дозволяють наочно відобразити необхідність синхронізації, послідовності у виконанні різних робіт. Віхи, як і інші кордону на діаграмі, НЕ є календарними місті. Зрушення віхи призводить до зрушення всього проекту. Тому діаграма Ганта не є, строго кажучи, графіком робіт. Крім того, діаграма Ганта не відображає значущості або ресурс ємності робіт, а не відображає суті робіт (області дії). Для великих проектів діаграма Ганта стає надмірно великоваговою і втрачає будь-яку наочність.

Зазначені вище недоліки і обмеження серйозно обмежують сферу застосування діаграми. Тим не менш, у даний час діаграма Ганта є стандартом де-факто в теорії і практиці управління проектами, по крайній мірі, для відображення Структури переліку робіт за проектом [7].

Діаграми Ганта була використана для планування графіку виконання магістерської дисертації.

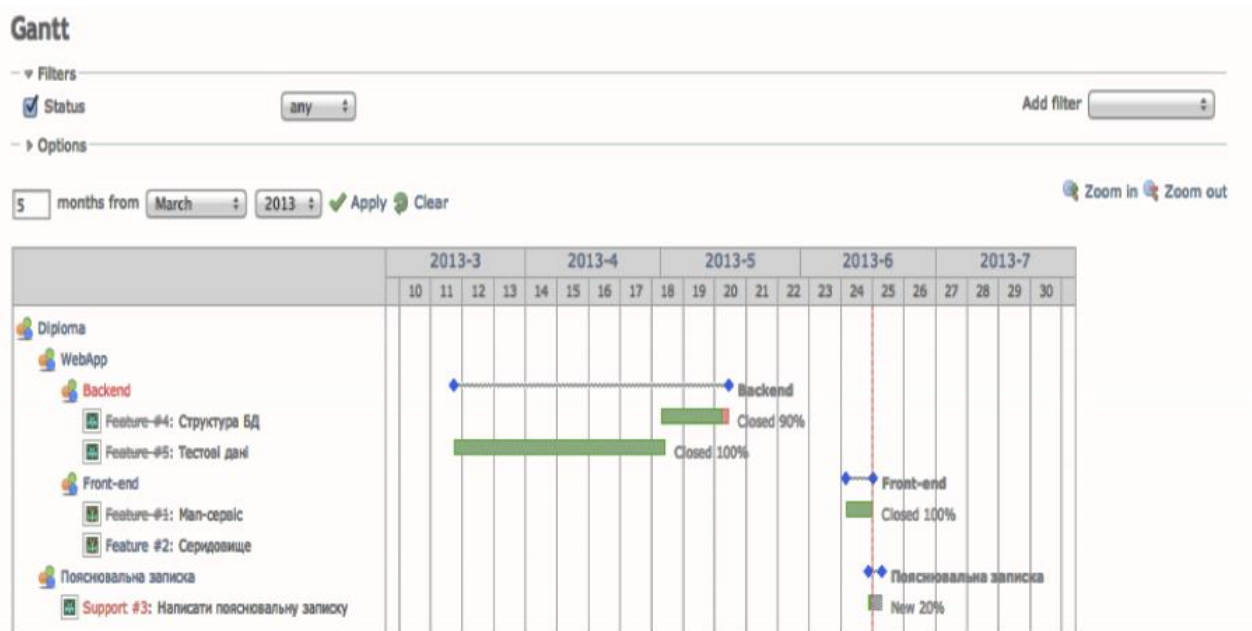


Рис 2.6 – Діаграма Ганта [7]

2.6 ВИСНОВКИ ДО РОЗДІЛУ

Порівняємо JADE та AnyLogic як основні інструменти розробки мультиагентної системи.

Таблиця 3.1 Порівняння інструментів розробки мультиагентних систем AnyLogic та JADE

	Anylogic	JADE
Ліцензія	Безкоштовний для користування в навчальних цілях	Безкоштовний, GNU
IDE	Графічний інтерфейс побудови логічних моделей та редактор	Будь яка IDE яка має підтримку Java
Документація	Наявна, велика кількість навчальних матеріалів та реалізацій типових задач	Наявна
Трудомісткість на реалізацію спроектованої моделі	Низька, використовуючи графічний інтерфейс	Висока
Візуалізація результатів моделювання	Наявна	Відсутня

Порівнявши процеси розробки мультиагентних систем в середовищах AnyLogic та JADE, прийнято рішення обрати AnyLogic. AnyLogic є безкоштовним для використання в навчальних або наукових сферах та розповсюджується за GNU ліцензією. Слід додати, що в цьому програмному забезпеченні є значний недолік перед конкурентом – це менша продуктивність створеного програмного забезпечення. Проте наявні переваги, такі як наявність графічного інтерфейсу для

побудови логіки моделей, якісна документація, що значно пришвидшить дослідження, переважають недоліки.

Велика кількість практичних задач при виконанні дисертації призводить до рішення розгорнути середовище управління проектом (Redmine). За допомогою використання репозиторію (GIT) було додано можливість працювати системою контролю версій, працюючи над розробкою агентної системи ітеративно та забезпечено можливість поверненню до попередньої версії в разі необхідності. Також було розгорнуто середовище виконання програмного забезпечення в хмарі, для надання можливості виконання роботи з різних робочих[7].

3 ПРИЙНЯТТЯ КОЛЕКТИВНИХ РІШЕНЬ АГЕНТАМИ В МУЛЬТИАГЕНТНИХ СИСТЕМАХ

Фундаментальним завданням багатьох системних мультиагентних мереж є успішне прийняття колективних рішень серед альтернатив за допомогою інформації, що поширюється по всій мережі. Групи окремих агентів, у програмах, включаючи транспортні та мобільні системи зондування, енерго та синтетичні біологічні мережі, часто вимагають єдиного вибору серед альтернатив, вибір істинного варіанта, як вчиняти та обрати необхідний керунок, або у випадку змін в середовищі та системі.

З метою розробки розподілених мультиагентних рішень ми прагнемо використовувати механізми, що використовуються групами тварин, виживання яких залежить від успішних колективних рішень серед наявних альтернатив. Домашні вуликові бджоли [66], зграйна риба [53] та перелітні птахи [54] приймають ефективні рішення, незалежно від конфліктів чи значних змін у середовищі. Бджоли використовують децентралізований підхід та стикаються з обмеженням відчуттів, комунікації та обчислення [60, 68], проте вони все ще діють із сталою швидкістю, точністю, надійністю та адаптивністю [64].

Типові механізми вивчення колективної поведінки тварин залежать від соціальної взаємодії тварин та їх сприйняття зовнішнього середовища. Точне розуміння цих залежностей робить можливим переведення механізмів у систематичну методологію біопроекування для використання в інженерних мережах. Однак, ще залишається викликом, зокрема, тому, що більшість досліджень колективної поведінки тварин ґрунтуються на емпіричній основі або залежать від моделей середнього поля.

Для вирішення цього завдання ми розглядаємо загальну агентно-орієнтовану динамічну модель прийняття рішень розподіленого між двома альтернативами. У цьому типі прийняття рішень виборча біфуркація є повсюдною [61]; це виявляється,

наприклад, у динаміці прийняття рішень домашніх медоносних бджіл та зграї золотого синця між джерелами поживи. Наш підхід полягає у введенні моделі агенту так, щоб вона також містила біфуркацію типу вилка. Це дозволяє тісно пов'язати динаміку групи тварин та мультиагентну динаміку шляхом відображення на звичайну форму біфуркації вилки.

Основний внесок цієї роботи полягає в наступному. По-перше, ми представляємо загальну агентну модель для біодинаміки колективного прийняття рішень, а також редукцію моделі та асимптотичне розширення, щоб показати, як модель фіксує адаптивні та стійкі ознаки динаміки у прийнятті рішень серед медоносних бджіл. Примітно, що медоносні бджоли надійно вибирають у користь найбільшого гнізда, а в разі альтернатив рівної цінності вони швидко роблять довільний вибір, якщо значення є досить високим. Дана динаміка медоносів була досліджена в роботах [63, 65, 66, 67].

По-друге, досліджено як значення альтернатив, індивідуальних переваг та топології взаємодії впливає на динаміку прийняття рішень. Ми мотивовані проблемою проектування колективної динаміки прийняття рішень, оскільки ці показники можуть служити контрольними параметрами в інженерно-технічних системах.

У пункті 3.2 запропоновано агентно-орієнтовану динамічну модель прийняття рішень. В 3.3 описано динаміку медоносних бджіл, де більше висвітлена адаптованість бджіл за допомогою прийняття важливих рішень. У пункті 3.4 представлений спосіб зменшення моделі до маловимірної придатної множини. Наведена модель використовується в п'ятому розділі, щоб показати, як модель відновлює значення динаміки важливих рішень медоносних бджіл, а висновки показують вплив інших параметрів системи.

3.1 МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ АГЕНТІВ

Представлена модель є спеціалізацією динаміки мережі Хопфілда [58, 59]. Модель забезпечує загальну динаміку прийняття рішень у мережі для набору N

взаємопов'язаних агентів і за дизайном демонструє біфуркацію типу вилка. Щоб описати прийняття рішень між двома альтернативами А і В, нехай $x_i \in \mathbb{R}$ - це стан агента i , що виражає його думку, з $i \in \{1, \dots, N\}$. Агент i має надати перевагу альтернативі А (В), якщо $x_i > 0$ (або < 0), з силою думки агента i , яку дає $|x_i|$. Якщо $x_i = 0$, агент i не визначено або не узгоджено.

Мережеві взаємозв'язки визначають, які агенти можуть виміряти стан іншого агента, і це показано за допомогою матриці мережевої суміжності $A \in \mathbb{R}^{N \times N}$. Кожен $a_{ij} \geq 0$ для $i, j \in (1, \dots, N)$ та $i \neq j$ дає вагу, що показує наскільки i -ий агент цінить вимірювання агента j . Тоді з $a_{ij} > 0$ випливає з того, що j є сусідом i . Ми встановимо $a_{ij} = 0$ для всіх i та $D \in \mathbb{R}^{N \times N}$ - буде діагональною матрицею з діагональними елементами $d_i = \sum_{j=1}^N a_{ij}$. $L = D - A$ - матриця графіка Лапласа, пов'язана з мережею взаємодії.

Визначаємо зміну думки кожного агента з плином часу як функцію від поточного стану агента, стана сусідів і можливий зовнішній стимул β_i :

$$\dot{x}_i = d_i x_i + \sum_{j=1}^N u a_{ij} S(x_j) + \beta_i \quad (1)$$

Кожен $\beta_i \in (\beta_A, 0 - \beta_B)$, $\beta_A, \beta_B \in \mathbb{R}^+$ описує зовнішній стимул, отриманий кожним агентом i , також може розглядатися як вподобання агента серед альтернатив. $\beta_i = \beta_A$ означає, що агент надає перевагу варіанту А, $\beta_i = \beta_B$ означає, що агент i надає перевагу варіанту В, а $\beta_i = 0$ означає, що агент не має ніяких уподобань. $u \geq 0$ - невід'ємний контрольний параметр, а $S: \mathbb{R} \rightarrow \mathbb{R}$ - гладка сигмоподібна функція, яка задовольняє наступні умови: $S'(z) > 0 \quad \forall z \in \mathbb{R}$ (монотонна); $S(z)$ належить сектору $(0, 1]$, а $\text{sgn } S''(z) = -\text{sgn } (z)$, де $(\cdot)'$ позначає похідну по аргументу функції.

Термін $uS(x_j)$ можна інтерпретувати як думку агента j , сприйнятого агентом i . $S(x)$ - це функція насичення, яка зменшує вплив більш вагомішої думки. Контрольний параметр визначає масштабування функції, тому можна розглядати його як зграєву роботу; більше уваги приділяється думці інших.

Нехай $x = (x_1, \dots, x_N)^T$, $\beta = (\beta_1, \dots, \beta_N)^T$ та $S(x) \in \mathbb{R}^N$ - вектор з елементами $S(x_i)$. Тоді (1) можна записати у векторній формі:

$$\dot{x} = -Dx + uAS(x) + \beta \quad (2)$$

Щоб прослідкувати динаміку біфуркації вилки, графік взаємозв'язку повинен перетинатись і бути зафіксований. Тоді $\text{rank}(L) = N - 1$ і $L1_N = 0$, де 1_N - це вектор N -стовпця з унітарними даними графі. L має нульове власне значення з відповідним власним вектором $x = \zeta 1_N$, $\zeta \in \mathbb{R}$, а кожне інше власне значення має позитивну дійсну частину. Зверніть увагу, що лінеаризація (2) при $x = 0$ для $u = 1$ і $\beta = 0$ - лінійна динаміка консенсусу $\dot{x} = -Lx$, яка сходиться до консенсусу $x = \zeta 1_N$. Це передбачає можливість біфуркації з центральним множинним тангенсом, до консенсусної множини [57, теорема 3.2.1]. За непарної симетрії (1) при $\beta = 0$ прийме форму вилки [55, теорема VI.5.1]. Це проілюстровано в теоремах 1 і рис. 3.1 для цілої мережі при $\beta = 0$:

$$\dot{x}_i = -(N - 1)x_i + \sum_{j=1, j \neq i}^N uS(x_i) \quad (3)$$

Теорема 1. Для постійності інваріантних множин динаміки (3) дотримуються наступні твердження:

- (i) Результуюча множина є глобально експоненціально постійною для кожного $u \in \mathbb{R} \geq 0$;
- (ii) $x = 0$ є глобально експоненціально постійним для $u \in [0, 1)$ і глобально асимптотично стійка для $u = u^* := 1$;
- (iii) $x = 0$ непостійна і існує дві постійні точки рівноваги на результуючій множині при $u > 1$.

Доведення. Починаючи з (i); розглянемо функцію Ляпунова

$$V_{ij}(x) = \frac{(x_i - x_j)^2}{2}. \text{ З цього випливає, що}$$

$$\dot{V}_{ij}(x) = -(N - 1)(x_i - x_j)(x_i - x_j + u(S(x_i) - S(x_j)))$$

$< -(N-1)(x_i - x_j)^2 = -2(N-1)V_{ij}$, для кожного $x_i \neq x_j$. Тому, для $V(x) = \sum_{i=1}^n \sum_{j=1}^n V_{ij}(x)$, $\dot{V}(x) < -2(N-1)V(x)$, для кожного $x \neq \zeta 1_N$, $\zeta \in \mathbb{R}$. $\dot{V}(x) = 0$ для $x_i = x_j = \zeta$, тому за принципом інваріантності Ласалле результуюча множина глобально експоненціально постійною. Використовуючи (i), достатньо вивчити динаміку (3) на консенсусній множині, де при $y = \frac{1}{N} \sum_{i=1}^N x_i$, вони зводяться до скалярної динаміки $\dot{y} = -(N-1)y + u(N-1)S(y)$.

(ii) та (iii) слідують шляхом перевірки цієї динаміки.

Вилка залишається для $\beta \neq 0$, коли симетрія системи зберігається, тобто коли $\beta_i = -\beta_A$ і кількість агентів з перевагою А дорівнюють кількості агентів з перевагою В. Доказом цього, є розширення до більш загальних випадків, а результати несиметричного випадку залежать від теорії сингулярності [55]. Асиметрія в системі призводить до розгортання біфуркації вилки, і це розгортання впливає на динаміку прийняття рішень, організовану вилками, і надійність, що виявляється при прийнятті рішень у медоносних бджіл.

3.2 БДЖОЛИНИЙ ВУЛИК ТА ЗНАЧЕННЯ ЗНАЧЕННЄВОГО ПРИЙНЯТТЯ РІШЕНЬ

Коли колонія медоносів сягає надто великих розмірів для свого вулика, частина колонії повинна відправитись на пошуки нового гнізда.

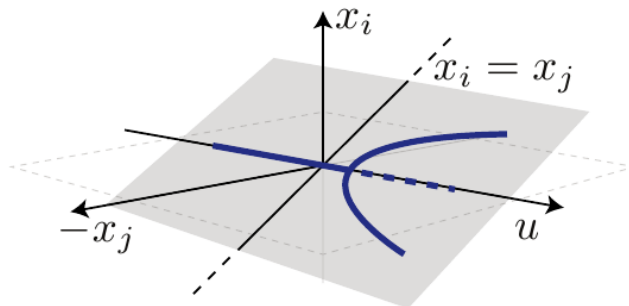


Рисунок 3.1 – Для $u = 1$ динаміка (3) виявляє біфуркацію вилки на $x = 0$. Постійні стани, що виникають при сингулярності, що лежить на результуючій

множині $\{x_i = x_j \mid i, j \in \{1, \dots, N\}\}$, показано сірим кольором. Гілки постійних та нестійких рішень показані як тверді та пунктирні лінії, відповідно [3].

Новий вулик повинен бути високоякісним, щоб колонія бджіл змогла пережити в ньому наступну зиму, і вибір має бути зроблений швидко через обмежене постачання їжі. ‘Відбуваючі’ бджоли чекають у рої, поки бджоли-розвідники шукають і оцінюють потенційні майданчики для вулика. Кожен розвідник повертається до рою кілька разів, щоб оповістити та залучити інших на свій обраний майданчик. Колективне рішення рою для одного з альтернативних майданчиків приймається за наявності кворуму.

Значення місця пов'язане з його об'ємом, висотою над землею та розміром і розташуванням вхідної порожнини. Було показано, що медоносні рої швидко та точно вибирають найбільш цінні майданчики серед альтернатив на основі цих критеріїв в [66]. Пристосованість та надійність цього процесу є одними з функцій, які ми прагнемо зафіксувати із запропонованою агентно-орієнтованою моделлю. Помітно, що медоносні бджоли ефективно вибирають один з майданчиків, коли вони мають однакову або майже рівну цінність [68]. Рішення є чутливим до відносного та абсолютного значення доступних майданчиків [63], і таким чином цей процес відноситься до значеннєвого прийнятного рішення. Чутливість до абсолютної величини робить медоносів адаптованими до зміни навколишнього середовища: якщо два рівні або майже рівні майданчики мають високу цінність, бджоли-медоноси будуть самовільно вибирати один з невеликими зусиллями групи. Однак, якщо вони мають низьку цінність, вони утримаються від вибору, і в кінцевому рахунку виберуть один з них опісля значних групових зусиль.

Механізми, що пояснюють важливість прийняття рішень, були широко вивчені в [63], використовуючи модель середнього поля, представлену в [68], яка є добре змішаною моделлю населення. Модель середнього поля демонструє біфуркацію виду у випадку рівнозначних альтернатив, що має надзвичайно важливе значення для чудової поведінки щодо прийняття рішень медоносних бджіл. Однак

ми не можемо використовувати модель середнього поля для розробки розподілених стратегій управління або для вивчення впливу на динаміку топології мережі або розподілу переваг по всій групі. Наш підхід - зробити це за допомогою загальної агентно-орієнтованою моделі (2). Оскільки агентно-орієнтована модель (2) демонструє біфуркацію вилки, вона може бути строго пов'язана з моделлю середнього поля та її результатами. Далі ми можемо розглянути ціннісно-прийнятну динаміку прийняття рішень агентно-орієнтованої моделі з точки зору розподілених властивостей системи, і ми можемо використовувати її для розробки адаптивного та надійного мультиагентного рішення мережі.

В процесі прийняття рішень медоноси, як відомо, використовують два механізми зв'язку: "танець бджіл" щоб сповістити інших про знайдений нектар та "стоп-сигнал" для попередження небезпеки. В [68] показано, що стоп-сигнал використовується для зупинки танців та набору бджоли для конкуруючих майданчиків, що дозволяє бджолам розбити тупикові ситуації між нерівними альтернативами. Вони виводять модель середньої динаміки популяційного рівня, беручи до уваги велику загальну кількість популяції бджіл N . Модель описує еволюцію трьох видів популяцій: $y_A(t) = \frac{N_A(t)}{N}$, $y_B(t) = \frac{N_B(t)}{N}$, $y_U(t) = \frac{N_U(t)}{N}$, де N_A , N_B та N_U є підгрупами бджол, придатних для майданчиків А, В та непридатних бджіл відповідно. Оскільки $N_A + N_B + N_U = N_i$, отже, $y_A + y_B + y_U = 1$, достатньо вивчити еволюцію лише двох віддалених популяцій:

$$\begin{aligned}\frac{dy_A}{dt} &= \gamma_A y_U - y_A(\alpha_A - \rho_A y_U + \sigma_B y_B) \\ \frac{dy_B}{dt} &= \gamma_B y_U - y_B(\alpha_B - \rho_B y_U + \sigma_A y_A).\end{aligned}\tag{4}$$

Тут γ_i - швидкість відкриття та відданості розвідництву, α_i - швидкість спонтанного відмови, Тут ρ_i - швидкість вербування та σ_i – швидкість появи стоп-сигналу. Передбачається, що $\gamma_i = \rho_i = v_i$ та $\alpha_i = \frac{1}{v_i}$, де v_i - оцінена величина гнізда i .

Також, $\sigma_i = \sigma$, і тому рівно для всіх підгруп. Рішення про кворум досягається, коли y_A або y_B перетинає деяку порогову точку $\omega \in (0.5, 1]$.

У випадку з рівною альтернативою ($v_A = v_B = v$) існує критичне значення сили зупинки сигналу $\sigma^* = \frac{4v^3}{(v^2-1)^2}$. Якщо $\sigma < \sigma^*$, система має одну глобально-стійку рівновагу при $y_A = y_B$, тобто тупиковій ситуації або немає рішення. Для $\sigma > \sigma^*$ вирішення тупикової ситуації є нестійким і є дві постійні рівноваги, кожна з якої відповідає рішенням для однієї з двох варіантів. Це біфуркація вилки з параметром біфуркації σ та біфуркаційним значенням $\sigma = \sigma^*$. Критичне значення σ обернено залежить від значення v двох альтернатив, що дозволяє бджолам адаптуватися до їх середовища. Припустимо, що бджоли використовують фіксовану кількість стоп-сигналів σ . Потім, вибираючи між двома малоцінними альтернативами, вони залишаються в тупиковій зоні, ймовірно, чекають іншого кандидата на гніздо. Але якщо дві рівноцінні альтернативи мають високу цінність, вони будуть швидко вибрати одну довільно. Існує також ймовірність того, що бджоли можуть збільшити кількість стоп-сигналів з плином часу, коли стає очевидним, що не з'явиться кращих альтернатив.

На рисунку 3.2 показаний діапазон значень v та швидкості зупинки сигналу σ , для яких існує одне рішення (тупик) та три рішення (два постійних рішення та один нестійкий тупик), а також двовимірні симплекси, у яких розвивається динаміка. Крива між регіонами описує зворотний зв'язок між точками біфуркації σ^* і значенням v .

Для вивчення цієї динаміки для рівних альтернатив в агентній-орієнтованій моделі встановимо відповідність на рисунку (3.2) за допомогою зміни шкали часу $\tau = vt$, що дає

$$\frac{dx}{d\tau} = -\frac{1}{v}Dx + \mu AS(x) + v \quad (5)$$

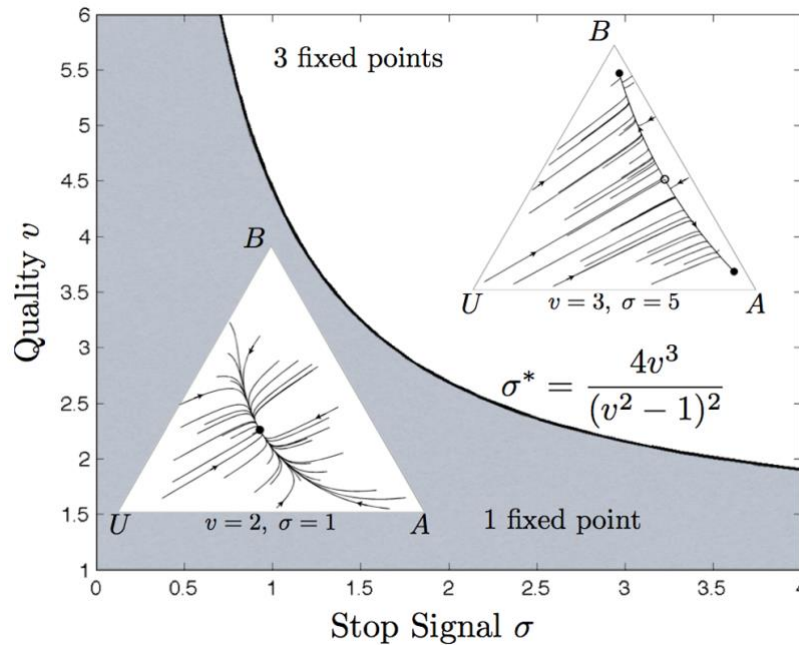


Рис 3.2 – 3 джерела [63]. Значення прийняття рішень для альтернатив з рівною величиною v у середній моделі (4). Симплекс ліворуч, представлений сірою зоною, показує збіг до єдиної стійкої рівноваги (чорного кола) в тупиковій зоні. Симплекс праворуч представника білої області, показує зближення двох постійних рівноваг при прийнятті рішення для A або B . Крива, що відокремлює область, описує зворотній зв'язок від σ^* до v .

Де $v = (v_1, \dots, v_N)^T$, $v_i \in (v, -v, 0)$ така, що $v = \sqrt{\beta}$, і $\mu = \frac{u}{v}$. Це відображення використовуватиметься у частині 3.5 для підключення результатів агенто-орієнтованої моделі до динаміки моделі середнього поля медоносів.

3.3 ЗНИЖЕННЯ МОДЕЛІ ДО НИЗЬКОГО РІВНЯ, ПРИЙНЯТНОЇ МНОЖИНИ

Повертаючись до агенто-орієнтованої моделі, для деяких класів мережевих графіків можна визначити загальноприйнятну низькорівневу множину, за допомогою яких можна зменшити динаміку (1), а також провести аналіз на зменшеній моделі. Розмірність N системи розглядається як дискретний параметр, що дозволяє вивчати чутливість динаміки до розмірів зафіксованих і незафіксованих популяцій.

Нехай n_1 та n_2 - кількість агентів з пріоритетом $\beta_i = \beta_A = \beta_1$ та $\beta_i = -\beta_B = \beta_2$ відповідно, а $n_3 = N - n_1 - n_2$ - це кількість агентів без будь-яких переваг $\beta_i = 0 = \beta_3$). З [10] розбиття вершин на клітини C_1, \dots, C_r називається рівноправним, якщо кожен вузол C_i має однакову кількість сусідів в C_1 для всіх i, j . Нехай $\phi_k \subset \{1, \dots, N\}$, $k \in \{1, 2, 3\}$ - індексний набір, пов'язаний з кожною з трьох груп n_1, n_2, n_3 такий, що $\phi_k \subset \{1, \dots, N\}$, $k \in \{1, 2, 3\}$ визначає рівний розподіл. Тоді ми можемо визначити динаміку думки кожного агента $i \in \phi_k$, $k \in (1, 2, 3)$, як

$$\dot{x}_i = -\bar{d}_k x_i + u \sum_{j \in \phi_k} S(x_j) + u \sum_{\substack{m \in \{1,2,3\} \\ m \neq k}} \sum_{j \in \phi_m} \bar{a}_{km} S(x_j) + \bar{\beta}_k, \quad (6)$$

для $i \in \phi_k$, де \bar{d}_k - в степені кожного агента в групі k , а \bar{a}_{km} - число сусідів, що кожен вузол групи k має групи m (включаючи $m = k$).

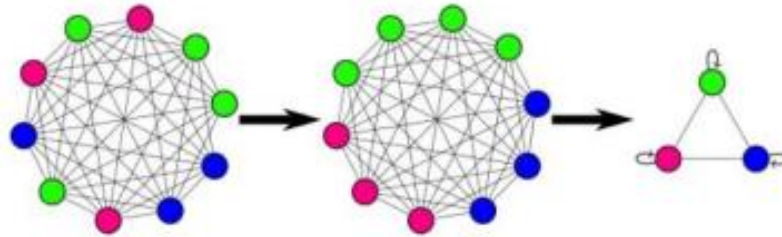


Рисунок 3.3 – Зменшення моделі для (2), визначеної на графічному алгоритмі з усіма різними графами з $N = 10$ вузлами (ліворуч), зменшується до 3-мірної системи, яка описується графіком справа. Агенти згруповані за інформаційною цінністю (середня), а думка в кожній групі сходяться до консенсусу відповідно до динаміки (7). Блакитні агенти мають $\beta_i = 1$, рожеві агенти $\beta_i = -1$ і зелені агенти $\beta_i = 0$. У зменшеній моделі агенти мають власну петельку, яка відображає вплив інших у тій самій групі.

Наступна теорема дозволяє аналізувати (6) обмеження на підпростір, де кожен агент у тій же групі має однакову думку. Центральна ідея теореми наведена на рисунку 3.3.

Теорема 2. Кожна траєкторія динаміки думки (6) експоненціально сходиться до тривимірної множини

$$\mathcal{E} = \{x \in \mathbb{R}^N \mid x_i = x_j, \forall i, j \in \phi_k, k = 1, 2, 3\}.$$

Динаміка на \mathcal{E}

$$\begin{aligned} \dot{y}_1 &= -\bar{d}_1 y_1 + u(n_1 - 1)S(y_1) + u(n_2 \bar{a}_{12} S(y_2) + n_3 \bar{a}_{13} S(y_3)) - \beta_A \\ \dot{y}_2 &= -\bar{d}_2 y_2 + u(n_2 - 1)S(y_2) + u(n_1 \bar{a}_{21} S(y_1) + n_3 \bar{a}_{23} S(y_3)) - \beta_B \\ \dot{y}_3 &= -\bar{d}_3 y_3 + u(n_3 - 1)S(y_3) + u(n_2 \bar{a}_{31} S(y_1) + n_2 \bar{a}_{32} S(y_2)). \end{aligned} \quad (7)$$

Доведення. Розглянемо функцію Ляпунова $V(x) = \sum_{k=1}^3 V_k(x)$, де $V_k(x) = \frac{1}{2} \sum_{i \in \phi_k} \sum_{j \in \phi_k} (x_i - x_j)^2$, для $k \in \{1, 2, 3\}$.

З цього слідує, що

$$\begin{aligned} V_k(x) &= \sum_{i \in \phi_k} \sum_{j \in \phi_k} (x_i - x_j)(\dot{x}_i - \dot{x}_j) \\ &= \sum_{i \in \phi_k} \sum_{j \in \phi_k} (-\bar{d}_k (x_i - x_j)^2 - u(x_i - x_j)(S(x_i) - S(x_j))) \\ &\leq -\bar{d}_k V_k(x), \end{aligned}$$

тому $\dot{V}(x) \leq -\bar{d}_k V(x)$. За принципом інваріантності Ласалле кожна траєкторія (6) експоненціально сходиться до найбільшої інваріантної множини в $\dot{V}(x) = 0$, що є різновидом \mathcal{E} . Нехай $y_k = x_1$, для будь-якого $i \in \phi_k, k \in \{1, 2, 3\}$. Тоді динаміка (6) скоротиться відразу до (7).

3.4 ОТРИМУВАННЯ СУМІСНОСТІ ТА ВПЛИВ СИСТЕМНИХ ПАРАМЕТРІВ

Векторне поле $F: \mathbb{R}^N \rightarrow \mathbb{R}^N$ - Z_2 -симетричне, коли воно переміщується з лінійним перетворенням

$$\gamma = \begin{bmatrix} 0_{n \times n} & -I_n & 0_{n \times (N-2n)} \\ -I_n & 0_{n \times n} & 0_{n \times (N-2n)} \\ 0_{(N-2n) \times n} & 0_{(N-2n) \times n} & -I_{N-2n} \end{bmatrix},$$

для деяких навіть $2_n < N$. Отже, динаміка реакції (2) Z_2 -симетрична, якщо $\beta_A = -\beta_B = \beta$ і $n_1 = n_2 = n$. Тобто, зворотний знак β_A та β_B еквівалентний застосуванню перетворення $x \rightarrow -x$. Розглядаючи (7) за симетрією Z_2 , можна знайти апроксимацію \hat{x}^* до точки біфуркації u^* .

3.4.1 НАБЛИЖЕННЯ ТОЧКИ БІФУРКАЦІЇ ДЛЯ СИМЕТРИЧНОЇ, ВСЕОСЯЖНОЇ МЕРЕЖІ Z_2

Згідно з вищесказаними припущеннями, а також з $n_3 = N - 2_n$ незадіяними агентами, динаміка (7) зменшується до

$$\begin{aligned} \dot{y}_1 &= -(N-1)y_1 + u((n-1)S(y_1) + n(S(y_2) + n_3S(y_3))) - \beta \\ \dot{y}_2 &= -(N-1)y_2 + u(nS(y_1) + (n-1)S(y_2) + n_3S(y_3)) - \beta \\ \dot{y}_3 &= -(N-1)y_3 + u(nS(y_1) + nS(y_2) + (n_3-1)S(y_3)). \end{aligned} \quad (8)$$

Тоді $y^* = (y^*, -y^*, 0)$ завжди в рівновазі, де y^* є рішенням

$$(N-1)y^* + uS(y^*) - \beta = 0. \quad (9)$$

Коли $\beta = 0$, то стан замкнутості $y = 0$ є рівновагою для (8) для всіх $u \in \mathbb{R}$. Якщо для $\beta \neq 0$, то під Z_2 -симетрією неявна функціональна теорема впливає з того, що тупикова множина плавно змінюється і що точка рівноваги $y^* = (y^*, -y^*, 0)$ де $y^* = y^*(u, \beta)$, така, що $y^*(u, 0) \equiv 0$, рівно залежить від u і β . Тоді можна знайти наближення до y^* .

Для прикладу, нехай $S(\cdot) = \tanh(\cdot)$ і починаємо з розкладу ряду Тейлора $y^*(u, \beta)$ по відношенню до β :

$$y^*(u, \beta) = \beta_{yI} + \beta_{yII}^2 + \beta_{yIII}^3 + \beta_{yIV}^4 + \delta(\beta^5).$$

Тоді замінюємо $y^*(u, \beta)$ на (9) і диференціюємо по відношенню до β , щоб отримати

$$(N-1)y^{*'}(u, \beta) + u \operatorname{sech}^2(y^*(u, \beta))y^{*'}(u, \beta) - 1 = 0.$$

Підставляючи $\beta = 0$, отримуємо $y_I = \frac{1}{N-1+u}$. Аналогічно для вищих порядків, дає $y_{II} = y_{IV} = 0$ и $y_{III} = \frac{u}{3(N-1+u)^4}$.

Тому

$$y^* = \frac{1}{N-1+u}\beta + \frac{u}{3(N-1+u)^4}\beta^3 + \delta(\beta^5).$$

На точці біфуркації u^* , Якоб'ян (8), обчислений при u^* спаді. Алгебраїчний вираз Якобіана, занадто довгий щоб відтворити тут, є функцією N , n_3 і u^* .

Нагадаємо, що u^* також є функцією β , тому, встановивши детермінант Якобіана, рівний нулю, отримаємо трансцендентне рівняння в u^* , яке можна вирішити обчислювально для u^* . Щоб наблизитись, почнемо з розкладу ряду Тейлора $u^*(\beta) = 1 + \beta_{u_1^*} + \beta_{u_2^*}^2 + \beta_{u_3^*}^3 + \delta(\beta^4)$. Продовжуючи відносно u^* вище, призводить до наступного $\delta(\beta^4)$ наближення до u^* :

$$\bar{u}^* = 1 + \frac{(1+3N^3)^2(N-n_3)}{9N^9}\beta^2. \quad (10)$$

Апроксимація (10) точки біфуркації u^* є функцією значень β , загальної розмірності групи N та розміром незадіяної групи n_3 ; це явно описує чутливість біфуркації до розміру групи та переваги сили.

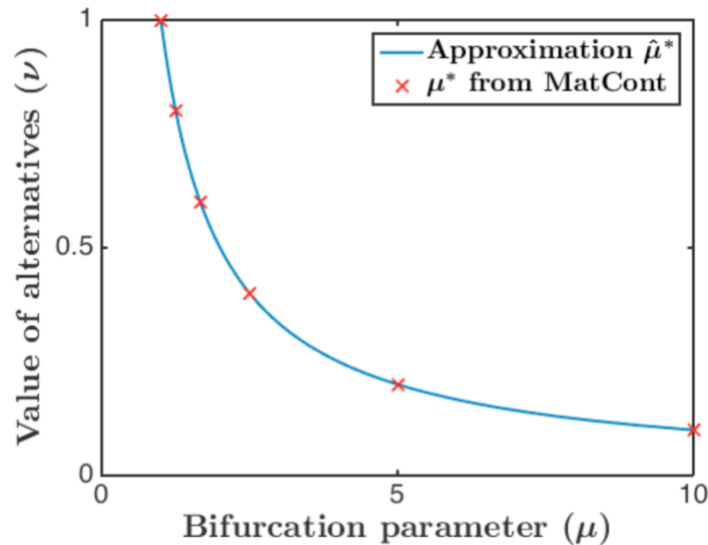


Рисунок 3.4 – Значення прийняття рішень для альтернатив з рівним значенням ν в агенто-орієнтованій моделі (5). Крива показує, як μ^* знаходиться у зворотній залежності від ν , відновлюючи значення чутливості моделі середнього поля медоносів (4); порівнюємо з рисунком 3.2. Синя лінія показує апроксимацію

μ^* (11), а на червоних хрестах показано μ^* що підраховано використовуючи програмне забезпечення для продовження діяльності. Розміри групи $n = n_3 = 30$ [3].

3.4.2 Відновлення значення динаміки медоносних бджіл

Для відновлення значення моделі середнього поля бджіл-медососів в агентно-орієнтованій моделі динаміки розглянемо динаміку у формі (5) для рівних альтернатив, де параметр біфуркації $\mu = \frac{u}{v}$ з $v = \sqrt{\beta}$. Застосування апроксимації (10) для u^* дає наближення до точки біфуркації μ^* для динаміки (5), оскільки

$$\hat{\mu}^* = \frac{1}{v} + \frac{(1+3N^3)^2(N-n_3)}{9N^9} v^3 + \delta(v^7) \quad (11)$$

На рисунку 3.4 показано, наскільки гарно $\hat{\mu}^*$ апроксимує μ^* , обчислено за допомогою програмного забезпечення MatCont. Як і у випадку моделі середнього-поля для бджіл-медоносів (див. рис. 3.2), точка біфуркації на основі агентно-орієнтованій моделі залежить від величини альтернатив v (див. [63] та рисунок 3.4). Таким чином, наша агентно-орієнтована модель прийняття рішень відновлює прийнятну для себе цінність прийняття рішень моделі середнього поля для бджіл-медососів. Ця цінність забезпечує ефективну та адаптовану динаміку прийняття рішень.

Залежність μ^* від v продемонстрована на рис. 3.5а, де діаграми біфуркації для агентно-орієнтованої моделі задані для діапазону значень v . Ми зауважимо, що точка біфуркації зменшується, оскільки v збільшується. Існує також збільшення різкості гілок біфуркації, оскільки v збільшується; це відповідає швидшому зростанню середньої думки.

3.4.3 Вплив параметрів системи.

Перевагою агентно-орієнтованої схеми є те, що вона дає змогу вивчати чутливість динаміки до системних параметрів, що варіюються у групі. Це буде розглянуто більш детально в наступній роботі, але деякі короткі результати представлені в цій роботі. На рис.3.5б - рис.3.5в показані ті ж відносини для $\hat{\mu}^*$ з

точки зору ν , як показано на рис.3.4, але з різним числом агентів та різними домовленостями підгрупи. Ефекти в цьому випадку невеликі, але існують певні тенденції [3].

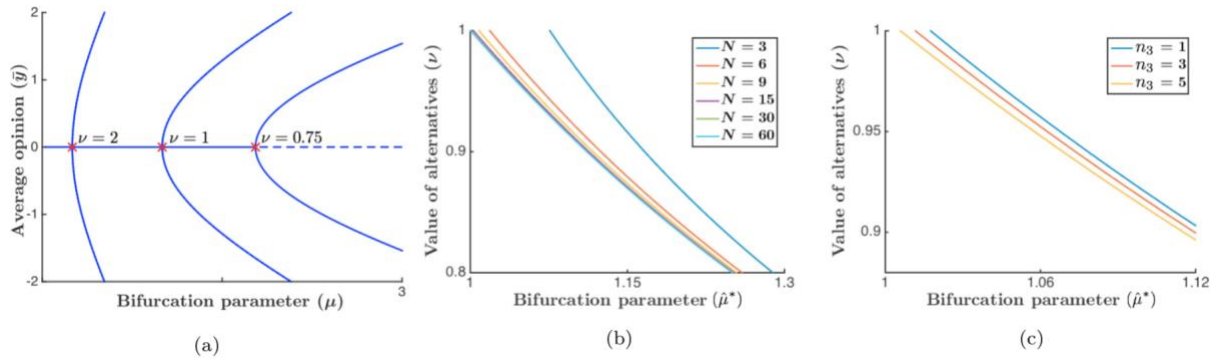


Рис 3.5 – (а) Діаграми біфуркації для агентно-орієнтованої моделі (5) з $n_3 = 20$, $N = 60$ та діапазоном ν , що показує залежність точки біфуркації та різкості на ν . (б) - (в) наближення до точки біфуркації $\hat{\mu}^*$ як функції ν для різних групових розмірів. (б) показує збільшення N з фіксованим $n_3 = \frac{N}{3}$, і (в) показує збільшення n_3 для фіксованого $N = 7$.

На рисунку 3.5б показано, як $\hat{\mu}^*$ зменшується з збільшенням загальної розмірності групи N , що означає, що для прийняття рішення більшою групою потрібні менші спільні зусилля. Також існує граничне значення $\mu^* = \frac{1}{\nu}$, яке швидко наближається до $N > 30$. З точки зору конструкції це показує, що якщо метою є мінімізація необхідних соціальних зусиль для групи агентів, то зменшується повернення при збільшенні розміру групи.

На рисунку 3.5в показано, що кількість соціальних зусиль також зменшується, коли кількість незадіяних агентів n_3 збільшується. В [53] показано, що кількість неінформованих агентів відіграє важливу роль у динаміці прийняття рішень зграї риб, тому цей результат дозволяє припустити, що агентно-орієнтована модель також зможе відобразити цю динаміку.

3.5 Висновки до розділу

Агентно-орієнтована модель прийняття рішень, про яку йдеться у даній роботі, пов'язує динаміку груп тварин та мультиагентних мереж. Вона утворює загальну структуру, що забезпечує досягнення характерних якостей прийняття рішень в колективі тварин (надійність, адаптивність) в інженерно-технічних мережесистемах. У найпростішому випадку "загальна комунікація" фіксує чутливість до прийняття рішень у медоносних бджіл. Попередні результати свідчать, що можливо зафіксувати демократичну динаміку консенсусу в групах медоносних бджіл.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ «МУЛЬТИАГЕНТНА СИСТЕМА ДЛЯ ПРИЙНЯТТЯ КОЛЕКТИВНИХ РІШЕНЬ АГЕНТАМИ»

Розділ має на меті проведення маркетингового аналізу стартап проекту “Мультиагентна система для прийняття колективних рішень” задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту “Мультиагентна система для прийняття колективних рішень” в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

4.1 ОПИС ІДЕЇ ТА ТЕХНОЛОГІЧНОЇ АУДИТ СТАРТАП-ПРОЕКТУ

Опис стартап-проекту “Мультиагентна система для прийняття колективних рішень” наведено у Таблиці 4.1.

Таблиця 4.1 – Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Ідея полягає у тому, щоб створити мультиагентну систему, завдяки якій можливо швидко виконувати операції по прийняттю колективних рішень у змінюваному середовищі.	1. Вирішення задачі по уникненню колізій	Розробникам ігор буде надана можливість використати готовий модуль для вирішення колізій для багатьох учасників. Наприклад для розрахунку руху натовпу людей без зіштовхувань.
	2. Вирішення задачі моніторингу стану здоров'я людей	Вендори зможуть використовувати запропоноване рішення в своїх системах моніторингу здоров'я людей. Система постійно автоматично приймає рішення для забезпечення найкращого стану людини, базуючись на інформації про середовище та інформацію з датчиків.

Отже, проект “Мультиагентна система для прийняття колективних рішень” може бути використаним як окремий модуль для вирішення задачі по уникненню колізій, так і бути системою моніторингу стану здоров’я людини у інфраструктурі інших додатків завдяки можливості підключення її як окремого модуля.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

No n/ n	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтра- льна сторона)	S (сильна сторон- а)
		Мій проект	Конк- урент1	Конкуре- нт2	Конку- рент3			
1.	Форма виконання	Окремий модуль	Про- грама	Веб- дода- ток	Про- грама			+
2.	Собівартість	Ни- зька	Ви- сока	Ни- зька	Ви- сока			+
3.	Наявність адміністра- тора для налаштування	Треба	Не треба, дистан- ційно	Треба	Треба		+	
4.	Наявність інтернету	Не треба	Необхі- дно	Не треба	Не треба			+
5.	Крос- платформен- ність	Ні	Так	Так	Ні	+		

Сильними сторонами проекту є форма виконання у вигляді окремого модулю, низька собівартість та можливість роботи без постійного підключення до інтернету. Слабкою стороною є необхідність людини-адміністратора для налаштування застосунку та те що система розробляється тільки для впровадження в певних системах. З цього можливо зробити висновок, що система є конкурентоспроможною.

Далі в таблиці 4.3 буде проведено аналіз технологічної здійсненності ідеї проекту.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

<i>No n/n</i>	<i>Ідея проекту</i>	<i>Технології її реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
1.	Створення мультиагентної системи для прийняття колективних рішень	JADE	Наявна	Безкоштовна, доступна
		AnyLogic	Наявна	Безкоштовна, доступна

Обрана технологія реалізації ідеї проекту: створення мультиагентної системи для прийняття колективних рішень за допомогою AnyLogic, так як не вимагає значних знань програмування, що призведе до зменшення загальної вартості на розробку.

4.2 АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

<i>No n/ n</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	200000 грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	$R = (2500000 * 100) / (700000 * 12) = 29,76\%$

Отже, було проаналізовано наявність попиту, обсяг, динаміку розвитку ринку. Обмеження для входу на ринок відсутні, динаміка ринку зростає, галузь є рентабельною.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

<i>No n/n</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1.	Необхідність автоматизувати процеси прийняття рішень в постійно змінюваних умовах	Потенційними цільовими групами є дослідницькі центри та компанії, специфіка роботи яких пов'язана із прийняттям колективних рішень	Цільова група займається дослідженнями або створює нові продукти	Рішення має бути швидким, що дозволить концентруватись на інших задачах, і використовувати даний програмний продукт як інструментарій

Згідно проведеної характеристики потенційних клієнтів стартап-проекту впливає, що на ринку є необхідність автоматизувати процеси прийняття рішень в постійно змінюваних умовах і потенційними цільовими групами є дослідницькі центри та компанії, специфіка роботи яких пов'язана із прийняттям колективних рішень.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. № 4.6-4.7). Фактори в таблиці подавати в порядку зменшення значущості.

Таблиця 4.6 – Фактори загроз

<i>No n/n</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
-------------------	---------------	----------------------	---------------------------------

1.	Конкуренція	Вихід на ринок великої компанії	1) Вихід з ринку 2) Запропонувати великій компанії поглинути себе 3) Передбачити додаткові переваги власного ПЗ для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок
2.	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	1) Передбачити можливість додавання нового функціоналу до створюваного ПЗ
3.	Надходження на ринок альтернативних продуктів	Перехід користувачів нашого товару на інший продукт	Впровадження нового функціоналу, якого немає у конкурентів

Отже, було проаналізовано фактори загроз ринкового впровадження проекту, серед яких: конкуренція, уповільнення росту ринку, зміна потреб користувачів та надходження на ринок альтернативних продуктів. Було також запропоновано можливі реакції компанії.

Таблиця 4.7 – Фактори можливостей

<i>No n/n</i>	<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанії</i>
1	Зростання можливостей потенційних покупців	Зростанням фінансування дослідницьких центрів у галузі охорони здоров'я	Запропонувати свої послуги і своє рішення дослідницьким центрам
2	Зниження довіри до конкурента 1	У ПЗ конкурента №1 нещодавно була знайдена вразливість, завдяки якій дані досліджень усіх клієнтів стали доступні в інтернеті для всіх користувачів	При виході на ринок звертати увагу покупців на безпеку нашого ПЗ

У Таблиці 4.7 наведено фактори можливостей ринкового впровадження проекту, серед яких: зростання можливостей потенційних покупців та зниження довіри до конкурента 1; було також запропоновано можливі реакції компанії.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Вказати тип конкуренції - досконала	Існує 3 фірми- конкурентки на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби - міжнародний	Всі компанії – з інших країн	Додати можливість вибору мови ПЗ, щоб бути одразу націленим на міжнародний ринок
3. За галузевою ознакою - внутрішньогалузева	Конкуренти мають ПЗ, яке використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко додати його як додатковий модуль
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме – програмне забезпечення	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг нецінова	Вдосконалення технології створення ПЗ, щоб собівартість була нижчою	Використання безкоштовних для розробки, на впровадження того що використовують конкуренти
6. За інтенсивністю - не марочна	Бренди відсутні	-

У Таблиці 4.8 наведено ступеневий аналіз конкуренції на ринку, де було визначено особливості конкурентного середовища та їх вплив а діяльність підприємства. Однією з найбільш важливих дій компанії для досягнення конкурентоспроможності є необхідність створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (Табл. 9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

<i>Складові аналізу</i>	<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
	<i>Навести перелік прямих конкурентів</i>	<i>Визначити бар'єри входження в ринок</i>	<i>Визначити фактори сили</i>	<i>Визначити фактори сили споживачів</i>	<i>Фактори загроз з боку замінників</i>

			<i>постачальників</i>		
Висновки:	Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 1, так як його рішення також представлене у вигляді програми.	Так, можливості для входу на ринок є, бо наше рішення спрощує та пришвидшує роботу спеціаліста.	Постачальники відсутні.	Важливим для користувача є швидкість роботи ПЗ.	Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару.

Було здійснено аналіз конкуренції в галузі за М. Портером, в результаті чого було визначено, що Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 1, так як його рішення також представлене у вигляді програми. Але можливості для входу на ринок є, бо наше рішення спрощує та пришвидшує роботу спеціаліста.

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також робиться висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку. Другий висновок враховується при формулюванні переліку факторів конкурентоспроможності у п. 4.6. На основі аналізу конкуренції, проведеного в п. 4.5 (табл. 4.9), а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. № 4.6-4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

<i>№ п/п</i>	<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
------------------	--	--

1.	Виконання ПЗ у вигляді окремого модулю	Це рішення дозволяє швидко додати розроблений функціонал до будь-якого проекту
2.	Можливість використання для моніторингу стану здоров'я	Користувач має лише завантажити дані і запустити програму на виконання.

У Таблиці 4.10 наведено обґрунтування факторів конкурентоспроможності, серед яких: виконання ПЗ у вигляді окремого модулю, що дозволить додати швидко додати розроблений функціонал до будь-якого проекту та можливість використання для моніторингу стану здоров'я, користування якого дуже просте. Користувач має лише завантажити дані і запустити програму на виконання.

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

У наступній таблиці наведено проведення аналізу сильних та слабких сторін стартап-проекту, факторами конкурентоспроможності виступили такі: виконання ПЗ у вигляді окремого модулю, можливість використання для моніторингу стану здоров'я.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

No n/n	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	+1	+2	+3
1	Виконання ПЗ у вигляді окремого модулю	20		+					
2	Можливість використання для моніторингу стану здоров'я	14	+						

Отже, сильною стороною проекту є виконання ПЗ у вигляді окремого модулю, що протиставляється можливості використання застосунку для моніторингу стану здоров'я. З наявних результатів порівняльного аналізу можливо виділити виконання у вигляді окремого модулю як найсильнішою стороною.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити 103 прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

У наступній таблиці буде проілюстровано SWOT-аналіз стартап-проекту, тобто його слабкі та сильні сторони, можливості та загрози виходу на ринок.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: можливість додати до проекту як окремий модуль, використання в сфері охорони здоров'я	Слабкі сторони: необхідність великої обчислюваної потужності
Можливості: у конкурента 1 виявлена проблема із безпекою ПЗ, додаткове фінансування для досліджень у підприємствах, які є потенційними покупцями	Загрози: конкуренція, зміна потреб користувачів

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

<i>№ n/n</i>	<i>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</i>	<i>Ймовірність отримання ресурсів</i>	<i>Строки реалізації</i>
1.	Створення монолітної програми яка надає свої обчислювані можливості	30%	10 місяців
2.	Створення програми яка є окремим модулем	80%	7 місяців

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими. Тому обираємо альтернативу створення програмного забезпечення яке є окремим модулем.

4.3 РОЗРОБКА РИНКОВОЇ СТРАТЕГІЇ ПРОЕКТУ

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

<i>No n/ n</i>	<i>Опис профілю цільової групи потенцій них клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1.	Університетські дослідження	Час виконання не є критичним у даному випадку, а однією із найголовніших переваг є саме це	Дослідження в області автоматичного прийняття колективних рішень проводяться постійно	Існує 3 конкуренти, які надають схожі, але менш зручні у використанні рішення.	У сегмент увійти не просто, бо бюрократія всередині університеті в занадто складна, також відсутня дуже впливова перевага ПЗ
2.	Дослідницькі центри	Пришвидшення часу виконання додає їм можливості виконати більшу кількість замовлень	Дослідження в області автоматичного прийняття колективних рішень і охорони здоров'я проводяться постійно, до того ж – за замовленням приватних компаній		Маючи перевагу у модульності і легкості використанні, вийти на ринок нескладно

3.	Підприємства	Автоматизація прийняття деяких колективних рішень додає їм можливості виконати більшу кількість замовлень			Маючи перевагу у тому, що рішення є зручним для користування і швидким вийти на ринок не є складно
Які цільові групи обрано: обираємо дослідницькі центри та підприємства, які підтримуються приватними компаніями					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку. Для роботи в обраних сегментах ринку необхідно сформуванню базову стратегію розвитку. За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Отже, проілюструвати базову стратегію розвитку можна у вигляді Таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвитку

<i>No n/ n</i>	<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку*</i>
1.	Створення програми яка є окремим модулем	Ринкове позиціонування	Модульність, можливість охопити галузь охорони здоров'я	Диференціації

Було обрано таку альтернативу розвитку проекту: Створення програми яка є окремим модулем за допомогою AnyLogic. Адже завдяки цьому рішення можна досягнути ключових конкурентоспроможних позицій кінцевого продукту.

Далі визначимо базову стратегію конкурентної поведінки в табл. 4.16

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

<i>No n/n</i>	<i>Чи є проект «періопрохід цем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки*</i>
-------------------	--	--	--	--

1.	Ні	Так	Буде, а саме: модульність, (конкурент 2)	Зайняття конкурентної ніші
----	----	-----	--	----------------------------

Отже, було визначено базову стратегію конкурентної поведінки як зайняття конкурентної ніші.

Визначимо стратегію позиціонування у Таблиці 4.17, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 – Визначення стратегії позиціонування

<i>No n/ n</i>	<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)</i>
1.	Простота у використанні, модульність	Диференціації	Простота у використанні, що дозволяє пришвидшити та спростити роботу працівників, модульність, що дозволяє використовувати як вже готове повноцінне рішення	Простота, модульність

Отже, було вибрано такі асоціації, які мають сформувати комплексну позицію власного проекту: простота у використанні (дозволяє пришвидшити та спростити роботу працівників) та модульність (дозволяє підключити до свого проекту і використовувати як вже готове повноцінне рішення).

4.4 РОЗРОБКА МАРКЕТИНГОВОЇ ПРОГРАМИ

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

<i>No n/n</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i>
1.	Простота у використанні	Просту документацію и легку можливість почати працювати з продуктів	Простота у використанні, що дозволяє легко навчити роботу працівників

2.	Модульність	Легко додати до проекту	Дозволяє використовувати як вже готове повноцінне рішення для вирішення своїх задач
----	-------------	-------------------------	---

Отже бачимо, що проект має ключові переваги перед конкурентами, які повністю відповідають потребам цільової аудиторії. Додаток реалізований у вигляді окремого модулю і може бути легко підключений до існуючих чи нових проектів, як готове рішення поставленої проблеми. Далі змістовна і водночас проста для розуміння документація дозволить легко навчити нових працівників використовувати дану розробку.

Далі у Таблиці 4.19 проілюстрована трирівнева маркетингова модель товару: уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Об'єкт допомагає підприємствам автоматизувати прийняття деяких колективних рішень. Компанія має лише додати до своєї системи готовий модуль та запустити його.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Простота інтерфейсу користувача 2. Модульність 3. Безпека згідно до світових стандартів	Нематеріальна	Технологічна
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє.		
	Моя компанія. «МультиАгент»		
III. Товар із підкріпленням	1-місячна пробна безкоштовна версія та безкоштовне встановлення		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: ноу-хау.			

Було описано три рівні моделі товару, з чого можна зробити висновок, що основні властивості товару у реальному виконанні є нематеріальними та технологічними. Також було надано сутність та складові товару у задумці та товару з підкріпленням.

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. У даному випадку найбільш вірогідним гарантом буде ноу-хау розробленої системи.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20 – Визначення меж встановлення ціни

<i>No n/n</i>	<i>Рівень цін на товари- замінники</i>	<i>Рівень цін на товари- аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
1.	200000	320000	500000	200000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21).

Таблиця 4.21 – Формування системи збуту

<i>No n/n</i>	<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
1.	Купують ПЗ та роблять щорічні внески для подовження ліцензії	Продаж	0(напрямую), 1(через одного посередника)	Власна та через посередників

Отже, система приносить прибуток завдяки щомісячним внескам для подовження ліцензії та придбанням підписок, продаж буде виконуватись напряму або через одного посередника.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

<i>No п/п</i>	<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікації, якими користують ся цільові клієнти</i>	<i>Ключові позиції, обрані для позиціонуван ня</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
1.	Купівля ПЗ через інтернет, робота з ПЗ на комп'ютерах з різними ОС	Інтернет	Швидкодія, простота використання, безпека	Показати переваги ПЗ, у тому числі і перед конкурентами	Демо-ролик із використання

Отже, в Таблиці 4.22 наведено концепцію маркетингових комунікацій, було визначено, що придбання ліцензії на користування буде здійснюватися в мережі Інтернет, необхідним буде щомісячне її продовження, користування ПЗ можливе на власних комп'ютерах з різними ОС.

4.6 ВИСНОВКИ ДО РОЗДІЛУ

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами.

Прийнято рішення, що стартап-проект “Мультиагентна система для прийняття колективних рішень” може бути використаним як окремий модуль для вирішення задачі по уникненню колізій, так і бути системою моніторингу стану здоров'я людини у інфраструктурі інших додатків завдяки можливості підключення її як окремого модуля. Також проект має ключові переваги перед конкурентами, які повністю відповідають потребам цільової аудиторії. Додаток реалізований у вигляді окремого модулю і може бути легко підключений до існуючих чи нових проектів, як готове рішення поставленої проблеми. Далі змістовна і водночас проста для розуміння документація дозволить легко навчити нових працівників використовувати дану розробку.

Для успішного виконання проекту необхідно реалізувати програму із використанням засобів AnyLogic. В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

ВИСНОВКИ

Було досліджено як агенти та мультиагентні системи використовуються для вивчення та моделювання складних систем у різних областях застосування, де фізичний фактор присутній для мінімізації енергії, коли фізичні об'єкти, як правило, досягають найнижчого споживання енергії у фізично стримуваному світі. Під час огляду літератури знайдено різні успішні застосування, де агенти та МАС були успішно експлуатовані, вивчення біологічних явищ, є синтез білків, загальне та актуальне явище в природі. У цій галузі кілька підходів для прогнозування тривимірної структури білків, наприклад, доступні в літературі Jennings et al. [32]. Деякі з них експлуатують хімічні чи фізичні властивості білків (наприклад, Standley et al. [44], Dudek та співавт. [25], робота з мінімізації енергії, тоді як Галактионов і Маршалл [28], Sabzevar et al. [40] використовує внутрішньоглобальні контакти); інші використовують еволюційну інформацію (Piccolbon and Mauri [37]). Деякі інструменти для прогнозування тривимірної структури білків були представлені в Douget і Labesse [27].

Показано, що фундаментальним завданням багатьох системних мультиагентних мереж є успішне прийняття колективних рішень серед альтернатив за допомогою інформації, що поширюється по всій мережі. Головним завданням групи окремих агентів, у програмах, включаючи транспортні та мобільні системи зондування, енерго та синтетичні біологічні мережі, є вибір єдиного рішення серед альтернатив, вибір істинного варіанта, як вчиняти та обрати необхідний керунок, або у випадку змін в середовищі та системі.

Для вирішення цього завдання було розглянуто загальну агентно-орієнтовану динамічну модель прийняття рішень розподіленого між двома альтернативами. У цьому типі прийняття рішень виборча біфуркація є повсюдною [61]; це виявляється, наприклад, у динаміці прийняття рішень домашніх медоносних бджіл та зграї золотого синця між джерелами поживи. Підхід полягає у введенні моделі агенту

так, щоб вона також містила біфуркацію типу вилка. Це дозволило тісно пов'язати динаміку групи тварин та мультиагентну динаміку шляхом відображення на звичайну форму біфуркації вилки.

Основний внесок цієї роботи полягає в наступному. По-перше, було представлено загальну агентну модель для біодинаміки колективного прийняття рішень, а також редукцію моделі та асимптотичне розширення, і показано, як модель фіксує адаптивні та стійкі ознаки динаміки у прийнятті рішень серед медоносних бджіл. Так як медоносні бджоли надійно вибирають у користь найбільшого гнізда, а в разі альтернатив рівної цінності вони швидко роблять довільний вибір, якщо значення є досить високим.

По-друге, досліджено як значення альтернатив, індивідуальних переваг та топології взаємодії впливає на динаміку прийняття рішень. Агентно-орієнтована модель прийняття рішень, про яку йдеться у даній роботі, пов'язує динаміку груп тварин та мультиагентних мереж. Вона утворює загальну структуру, що забезпечує досягнення характерних якостей прийняття рішень в колективі тварин (надійність, адаптивність) в інженерно-технічних мережевих системах. У найпростішому випадку "загальна комунікація" фіксує чутливість до прийняття рішень у перелітних птахів та медоносних бджіл. Попередні результати свідчать, що можливо зафіксувати демократичну динаміку консенсусу в зграях риб.

Також в рамках даного дослідження була запропонована реалізація стартап-проекту, розраховані основні фінансово-економічні показники та проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами. Для успішного виконання проекту необхідно реалізувати програму із використанням засобів AnyLogic.

ПЕРЕЛІК ПОСИЛАНЬ

1. Wooldridge, M., Ciancarini, P., 2001. Agent-oriented software engineering: The state of the art. Agent-Oriented Software Engineering. Springer Verlag. p. 1–28.
2. Wooldridge, M.J., 2009. Introduction to multiagent systems, second ed. Wiley.
3. Севідов П.М. Прийняття колективних рішень в мультиагентних системах / П. М. Севідов. // Міжнародний науковий журнал "Інтернаука". – 2018. – №7. – С. 81–101.
4. ИТ инфраструктура и телекоммуникации [Електронний ресурс]. – Режим доступу: <http://www.anylogic.ru/consulting/information-and-telecommunication-networks>. – Назва з екрану.
5. Системи управління версіями [Електронний ресурс]. – Режим доступу: <http://goo.gl/VBxMa>. – Назва з екрану.
6. Subversion [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/Subversion..> – Назва з екрану.
7. Управління проектами [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Управління_проектами. – Назва з екрану.
8. Діаграма Ганта [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Діаграма_Ганта. – Назва з екрану.
9. Smart Python Agent Development Environment [Електронний ресурс]. – Режим доступу: <https://github.com/javipalanca/spade>. – Назва з екрану.
10. Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood. Developing Multi-Agent Systems with JADE / Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood . Wiley, 2007. –P. 1-40. – ISBN: 978-0-470-05747-6
11. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A BDI reasoning engine. In Bordini et al. [5], chapter 6, pages 149–174.
12. AnyLogic [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/AnyLogic>. – Назва з екрану.

13. S. Railsback and V. Grimm. /Agent-based and Individual-based Modeling: A Practical Introduction. // Princeton University Press, 2011.
14. Odd Myklebust, Enterprise Modelling supported by Manufacturing Systems Theory: Doctoral dissertation, Norwegian University of Science and Technology Department of Production- and Quality Engineering.
15. Mark S. Fox and Michael Gruninger/ Enterprise Modeling //AI MAGAZINE, Fall 1998
16. Abar, S., Theodoropoulos, G.K., Lemarinier, P., OHare, G.M., 2017. Agent based modelling and simulation tools: A review of the state-of-art software. Computer Science Review 24, 13–33.
- 17.Allan, R., 2009. Survey of agent based modelling and simulation tools.
- 18.Bařdicař, C., Budimac, Z., Burkhard, H.D., Ivanovic, M., 2011. Software agents: Languages, tools, platforms. Computer Science and Information Systems 8, 255–298.
- 19.Baldoni, M., Baroglio, C., Mascardi, V., Omicini, A., Torroni, P., 2010. Agents, multi-agent systems and declarative programming: What, when, where, why, who, how?
- 20.Dovier, A., Pontelli, E. (Eds.), A 25-Year Perspective on Logic Programming: Achievements of the Italian Association for Logic Programming, GULP. Springer, pp. 204–230.
- 21.Bellifemine, F., Caire, G., Greenwood, D., 2007. Developing multi-agent systems with JADE. Wiley Series in Agent Technology. John Wiley & Sons.
- 22.Bergenti, F., Caire, G., Gotta, D., 2015. Large-scale network and service management with WANTS. Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier. pp. 231–246.
- 23.Bergenti, F., Gleizes, M.P., Zambonelli, F. (Eds.), 2004. Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook.

- 24.Boella, G., van der Torre, L.W.N., Verhagen, H., 2007. Introduction to normative multiagent systems. In: Boella, G., van der Torre, L.W.N., Verhagen, H. (Eds.), *Normative Multi-agent Systems*, 18–23, March 2007, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- 25.Bordini, R.H., Braubach, L., Dastani, M., et al., 2006. A survey of programming languages and platforms for multi-agent systems. *Informatica* 30.
- 26.Demazeau, Y., 1995. From interactions to collective behaviour in agent-based systems. In: *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo, pp. 117–132.
- 27.Douguet, D., Labesse, G., 2001. Easier threading through web-based comparisons and cross-validations. *Bioinformatics* 17, 752–753.
- 28.Dudek, M., Ramnarayan, K., Ponder, J., 1998. Protein structure prediction using a combination of sequence homology and global energy minimization: II. Energy functions.
- 29.*Journal of Computational Chemistry* 19, 548–573.
- 30.Dyson, G.B., 1997. *Darwin Among the Machines: The Evolution of Global Intelligence*. Addison-Wesley Publishing Company.
- 31.Galaktionov, S.G., Marshall, G.R., 1995. Properties of intraglobular contacts in proteins: An approach to prediction of tertiary structure, pp. 326–335.
- 32.Gough, J., Karplus, K., Hughey, R., Chothia, C., 2001. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *Journal of Molecular Biology* 313, 903–919.
- 33.Hewitt, C., Bishop, P., Steiger, R., 1973. A universal modular ACTOR formalism for artificial intelligence. In: Nilsson, N.J. (Ed.), *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, Standford, CA, August 1973, William Kaufmann, pp. 235–245.
- 34.Hollander, C.D., Wu, A.S., 2011. The current state of normative agent-based systems. *J. Artificial Societies and Social Simulation* 14.

35. Jennings, N., Sycara, K., Wooldridge, M., 1998. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1, 7–38.
36. Kardas, G., 2013. Model-driven development of multiagent systems: A survey and evaluation. *The Knowledge Engineering Review* 28, 479–503.
37. Kravari, K., Bassiliades, N., 2015. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation* 18, 11.
38. Meller, J., Elber, R., 2001. Linear programming optimization and a double statistical filter for protein threading protocols. *Proteins: Structure, Function and Genetics* 45, 241–261.
39. Omicini, A., Ricci, A., Viroli, M., 2008. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 17, 432–456. [Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems].
40. Piccolbon, A., Mauri, G., 1998. Application of evolutionary algorithms to protein folding prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1363, 123–135.
41. Rollings, A., Adams, E., 2003. Andrew Rollings and Ernest Adams on Game Design.
42. Russell, S.J., Norvig, P., 2003. *Artificial Intelligence: A Modern Approach*, second ed. Pearson Education.
43. Sabzekar, M., Naghibzadeh, M., Eghdami, M., Aydin, Z., 2017. Protein-sheet prediction using an efficient dynamic programming algorithm. *Computational Biology and Chemistry* 70, 142–155.
44. Savarimuthu, B., Purvis, M., Purvis, M., 2008. Social norm emergence in virtual agent societies, pp. 1485–1488.
45. Shoham, Y., 1993. Agent-oriented programming. *Artificial Intelligence* 60, 51–92.

46. Shoham, Y., Tennenholtz, M., 1992. On the synthesis of useful social laws for artificial agent societies (preliminary report), pp. 276–281.
47. Standley, D., Gunn, J., Friesner, R., McDermott, A., 1998. Tertiary structure prediction of mixed/proteins via energy minimization. *Proteins: Structure, Function and Genetics* 33, 240–252.
48. Verhagen, H., 2000. Norm autonomous agents.
49. Villatoro, D., 2011. Self-organization in decentralized agent societies through social norms, pp. 1297–1298.
50. Villatoro, D., Sen, S., Sabater-Mir, J., 2010. Of social norms and sanctioning: A game theoretical overview. *International Journal of Agent Technologies and Systems (IJATS)* 2, 115.
51. Zhang, Y., Leezer, J., 2009. Emergence of social norms in complex networks, pp. 549–555.
52. Zhengping, L., Cheng, H., Malcolm, Y., 2007. A survey of emergent behavior and its impacts in agent-based systems, pp. 1295–1300.
53. Couzin, I.D., Ioannou, C.C., Demirel, G., Gross, T., Torney, C.J., Hartnett, A., Conradt, L., Levin, S.A., and Leonard, N.E. (2011). Uninformed individuals promote democratic consensus in animal groups. *Science*, 334(6062), 1578–1580.
54. Eikenaar, C., Klinner, T., Szostek, L., and Bairlein, F. (2014). Migratory restlessness in captive individuals predicts actual departure in the wild. *Biology Letters*, 10(4).
55. Golubitsky, M. and Schaeffer, D.G. (1985). *Singularities and Groups in Bifurcation Theory*, volume 51 of *Applied Mathematical Sciences*. Springer-Verlag, New York, NY.
56. Govaerts, W. and Kuznetsov, Y.A. (2015). MATCONT and CL MATCONT: Continuation toolboxes in MATLAB. <https://sourceforge.net/projects/matcont/>.

57. Guckenheimer, J. and Holmes, P. (2002). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer, New-York, 7th edition.
58. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558.
59. Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of twostate neurons. *Proceedings of the National Academy of Sciences*, 81(10), 3088–3092.
60. Krause, J. and Ruxton, G.D. (2002). *Living in Groups*. Oxford University Press.
61. Leonard, N.E. (2014). Multi-agent system dynamics: Bifurcation and behavior of animal groups. *IFAC Annual Reviews in Control*, 38(2), 171–183.
62. Mesbahi, M. and Egerstedt, M. (2010). *Graph theoretic methods in multiagent networks*. Princeton Series in Applied Mathematics. Princeton University Press.
63. Pais, D., Hogan, P.M., Schlegel, T., Franks, N.R., Leonard, N.E., and Marshall, J.A.R. (2013). A mechanism for value-sensitive decision-making. *PloS ONE*, 8(9), e73216.
64. Parrish, J.K. and Edelstein-Keshet, L. (1999). Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411), 99–101.
65. Reina, A., Valentini, G., Fern´andez-Oto, C., Dorigo, M., and Trianni, V. (2015). A design pattern for decentralised decision making. *PLoS ONE*, 10(10), e0140950.
66. Seeley, T.D. and Buhrman, S.C. (2001). Nest-site selection in honey bees: how well do swarms implement the “best-of-N” decision rule *Behavioral Ecology and Sociobiology*, 49, 416–427.
67. Seeley, T.D., Visscher, P.K., Schlegel, T., Hogan, P.M., Franks, N.R., and Marshall, J.A.R. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science*, 335(6064), 108–111.
68. Sumpter, D.J.T. (2010). *Collective Animal Behavior*. Princeton University Press.